



Universidad
Carlos III de Madrid

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO FIN DE GRADO

Diseño de la telemetría para una motocicleta de competición I

MARZO 2015

AUTOR: ÁLVARO BLASCO PÉREZ

TUTOR: JUAN CARLOS GARCÍA PRADA

ÍNDICE

ÍNDICE

1. INTRODUCCIÓN	1
1.1 INTRODUCCIÓN	3
1.2 MOTIVACIONES	3
1.3 ¿QUÉ ES ARDUINO?.....	4
2. EXPLICACIÓN DEL FUNCIONAMIENTO DEL SISTEMA	9
2.1 OBJETIVOS	9
2.2 FASES DEL PROCESO	9
3. CAPTACIÓN DE DATOS	15
3.1 ARDUINO MEGA ADK	15
3.1.1 ENTORNO DE DESARROLLO ARDUINO	19
3.1.2 LIBRERÍAS DE ARDUINO.....	22
3.1.3 MONITORIZACIÓN SERIE	23
3.2 SENSOR DE TEMPERATURA.....	24
3.2.1 INTRODUCCIÓN	24
3.2.2 PRINCIPALES CARACTERÍSTICAS	26
3.3 ACELERÓMETRO	26
3.3.1 ¿QUÉ ES UN ACELERÓMETRO?	26
3.3.2 PRINCIPALES CARACTERÍSTICAS	27
3.4 MEDIDORES DE ELONGACIÓN.....	27
3.4.1 ¿EN QUÉ CONSISTEN?	27
3.4.2 PRINCIPALES CARACTERÍSTICAS	28
3.5 GPS	28
3.5.1 INTRODUCCIÓN	28
3.5.2 PRINCIPALES CARACTERÍSTICAS	30
3.5.3 FUNCIONAMIENTO.....	30
3.6 SD CARD SHIELD	33
3.6.1 INTRODUCCIÓN	33
3.6.2 PRINCIPALES CARACTERÍSTICAS	35
3.6.3 FUNCIONAMIENTO.....	35
4. VISUALIZACIÓN DE DATOS	38

5. SIMULACIÓN REAL.....	44
6. CONCLUSIONES	49
7. FUTUROS TRABAJOS.....	53
8. BIBLIOGRAFÍA	57
9. ANEXOS	61
9.1 ANEXO I: CÓDIGO	61
9.2 ANEXO II: DATASHEETS	74
9.2.1 SENSOR DE TEMPERATURA LM 35.....	74
9.2.2 ACCELERÓMETRO ADXL 335	88
9.2.3 VENUS GPS	104

Capítulo 1:

INTRODUCCIÓN

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

1. INTRODUCCIÓN

1.1 INTRODUCCIÓN

En el presente capítulo se hace una breve introducción general al contenido de este documento, las motivaciones a la hora de realizar el proyecto y los distintos objetivos del mismo.

En él se indica la causa que propició la creación del proyecto y, además, las pautas e ideas seguidas durante su desarrollo con el objetivo de optimizar el resultado final, de tal manera que éste fuera fiable y útil.

1.2 MOTIVACIONES

Este proyecto nace de la necesidad de complementar el trabajo del grupo MAQLAB (Grupo de Investigación Avanzado en Síntesis, Análisis, Modelado y Simulación de Máquinas y Mecanismos) y su trabajo de diseño y construcción de una motocicleta para la competición Motostudent.

Una vez llevado a cabo el diseño y la construcción de la parte mecánica de esta motocicleta, era necesario el desarrollo de un sistema de adquisición de datos capaz de aportar información práctica para la conducción. Incorporando una interfaz gráfica a este sistema se podría desarrollar un módulo de visualización práctico y eficaz, con un gran número de posibilidades tanto en circuito como en carretera.

Una de las principales premisas para la elaboración de este proyecto ha sido la utilización de elementos de bajo coste. Por ello, para gestionar todos los dispositivos necesarios, se ha empleado un microcontrolador Arduino, conocidos por su sencillez, facilidad para programar y su bajo precio. Además, otra de las ventajas que ofrece este microcontrolador es su tamaño. Al tratarse de un dispositivo de pequeñas dimensiones resulta ideal para la elaboración de un sistema de adquisición y visualización de datos compacto, lo que favorece su funcionalidad y facilita su posterior montaje en el chasis de la motocicleta.

Las placas Arduino son de software libre, hecho que aporta una mayor flexibilidad a la hora de añadir los distintos dispositivos necesarios para el sistema de adquisición de datos y ofrece una

mayor facilidad para realizar futuras ampliaciones del proyecto llevadas a cabo por el grupo MAQLAB.

1.3 ¿QUÉ ES ARDUINO?

Como ya he citado anteriormente, el microcontrolador utilizado es el Arduino. A continuación explicaré qué es Arduino.

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP o Pure Data. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador.

El proyecto Arduino recibió una mención honorífica en la categoría de Comunidades Digital en el Prix Ars Electrónica de 2006.



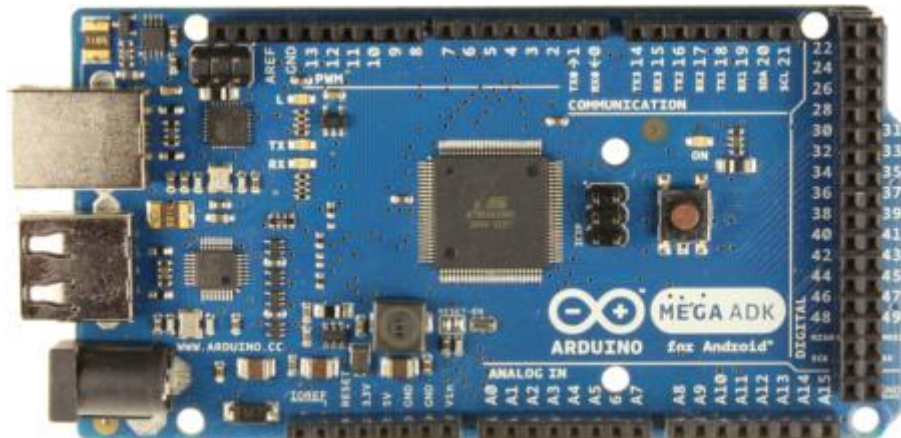
Ilustración 1: Logo de Arduino

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

En nuestro caso la placa elegida para realizar este proyecto es la Arduino Mega ADK. Sus características son las siguientes:

- **Microcontrolador:** ATmega2560
- **Tensión de funcionamiento:** 5 V
- **Voltaje de entrada (recomendado):** 7-12 V
- **Voltaje de entrada (límites):** 6-20 V
- **Pines digitales E/S:** 54 (de los cuales 15 proporcionan salida PWM)
- **Pines analógicos de entrada:** 16
- **Corriente DC por pin de E/S:** 40 mA
- **Corriente DC del pin de 3,3V:** 50 mA
- **Memoria Flash:** 256 KB de los cuales 8 KB son utilizados por el gestor de arranque
- **SRAM:** 8 KB
- **EEPROM:** 4 KB
- **Frecuencia de reloj:** 16 MHz
- **USB Host Chip:** MAX3421E



DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

Capítulo 2:

EXPLICACIÓN DEL FUNCIONAMIENTO DEL SISTEMA

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

2. EXPLICACIÓN DEL FUNCIONAMIENTO DEL SISTEMA

2.1 OBJETIVOS

En este proyecto se busca la implementación de un sistema de adquisición de datos para la motocicleta de MAQLAB basado en un microcontrolador de la plataforma de software libre Arduino y en una serie de dispositivos periféricos conectados a ella. Estos datos, además, serán guardados en una tarjeta SD para luego ser tratados y mostrados en una interfaz gráfica, lo que permitirá conocer ciertas variables de interés que serán útiles para el equipo durante la competición.

Estos objetivos pueden desglosarse en el siguiente guión de trabajo:

- Búsqueda bibliográfica de las diferentes técnicas utilizadas en sistemas de telemetría.
- Búsqueda bibliográfica de los distintos modelos de microcontroladores que sean capaces de cumplir con los requisitos para el desarrollo del proyecto.
- Búsqueda bibliográfica de las diferentes opciones a la hora de elegir los distintos sensores que conforman el sistema de adquisición de datos.
- Estudio y Realización del conexionado entre los diferentes elementos: Arduino, sensores, módulo de tarjeta SD, etc.
- Desarrollo del software que permita la comunicación entre los sensores, Arduino y módulo de tarjeta SD, de tal manera que todos los dispositivos interactúen entre sí.
- Simulaciones reales de recorridos en calle.

2.2 FASES DEL PROCESO

Una vez fijados los objetivos del proyecto, se seleccionaron cuáles serían los datos a recoger por el sistema. Estos deberían aportar información de interés al usuario durante la competición, por lo que se estableció que serían:

- Temperatura ambiente.
- Variables direccionales de aceleración.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

- Tiempo real.
- Coordenadas GPS.
- Elongación de suspensiones delantera y trasera.

Para ello serían cinco los sensores necesarios conectados al microcontrolador Arduino: un módulo GPS, un sensor de temperatura, un acelerómetro y dos potenciómetros que miden la elongación de las suspensiones.

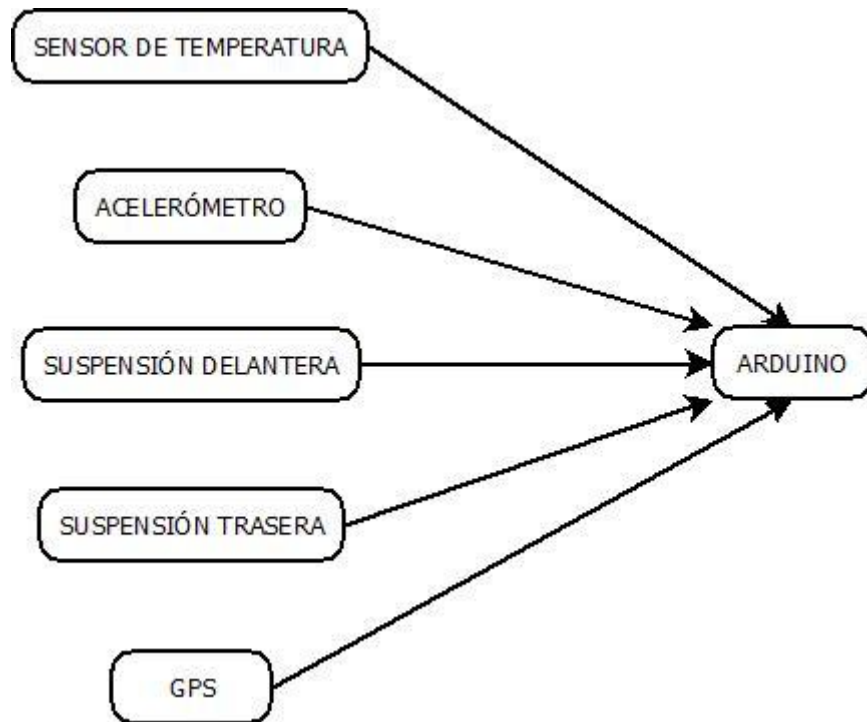


Ilustración 2: Esquema de sensores

Los datos recogidos por los distintos sensores, son guardados en una tarjeta SD que está insertada en un módulo conectado al Arduino.

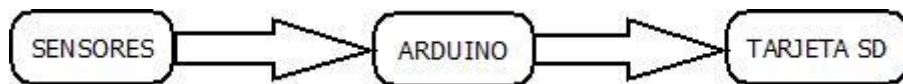


Ilustración 3: Esquema de captación de datos

Los datos escritos en la tarjeta SD sirven en la segunda parte de este proyecto para ser tratados y representados gráficamente y así poder sacar una lectura del recorrido en circuito o en la calle.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

La tarjeta SD será introducida a otro microcontrolador Arduino que se encarga de dicha función.



Ilustración 4: Esqueme de captación de datos

Capítulo 3:

CAPTACIÓN DE DATOS

3. CAPTACIÓN DE DATOS

En este capítulo se da información específica de todos los elementos que componen el módulo de adquisición de datos que irá instalado en la motocicleta. A continuación, se explicará qué elementos son, cómo funcionan, cuáles son sus principales características, por qué se eligieron y cómo han de ir conectados entre sí.

3.1 ARDUINO MEGA ADK

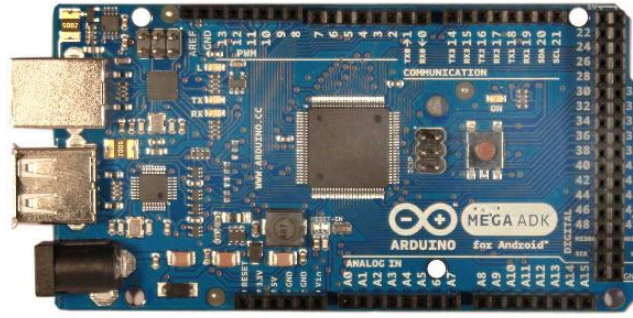
Anteriormente, ya expliqué en qué consistía el microcontrolador Arduino y en éste apartado daré detalles sobre el modelo que hemos utilizado que es el Arduino Mega ADK.

La idea inicial a la hora de comenzar el proyecto era la de usar la placa Arduino UNO, el modelo básico dentro de la gama de Arduino. Esta placa está basada en el microcontrolador ATmega328 y consta de 14 entradas/salidas digitales y 6 entradas analógicas. Tras establecer los objetivos finales del sistema de adquisición y visualización de datos, se optó por la elección de otro tipo de placa más potente debido a la gran carga de tareas que tendría que soportar.

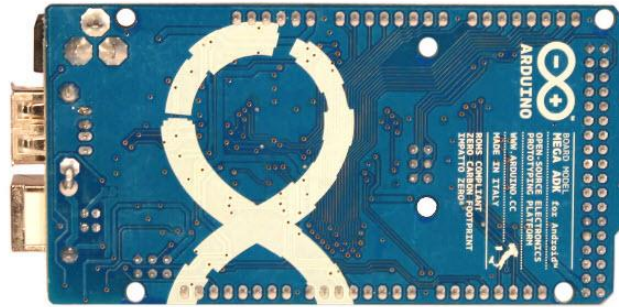
Finalmente la placa elegida fue la Arduino Mega ADK. Dado el número de elementos periféricos a utilizar y, sobre todo, pensando en posibles futuras ampliaciones del sistema, era necesario escoger un dispositivo con un número adecuado de entradas y salidas analógicas y digitales y con suficiente potencial para trabajar con fluidez.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez



Arduino ADK board front view



Arduino ADK board rear view

Ilustración 5: Placa Arduino Mega ADK

Las características de este modelo son las siguientes:

- **Microcontrolador:** ATmega2560
- **Tensión de funcionamiento:** 5 V
- **Voltaje de entrada (recomendado):** 7-12 V
- **Voltaje de entrada (límites):** 6-20 V
- **Pines digitales E/S:** 54 (de los cuales 15 proporcionan salida PWM)
- **Pines analógicos de entrada:** 16
- **Corriente DC por pin de E/S:** 40 mA
- **Corriente DC del pin de 3,3V:** 50 mA
- **Memoria Flash:** 256 KB de los cuales 8 KB son utilizados por el gestor de arranque
- **SRAM:** 8 KB
- **EEPROM:** 4 KB
- **Frecuencia de reloj:** 16 MHz
- **USB Host Chip:** MAX3421E

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

Cabe destacar el LED incluido en el pin digital 13, de gran ayuda cuando se comienza a usar el Arduino. Permite conocer si el programa se ha cargado y se ejecuta de forma correcta.

A pesar del gran número de pines del que dispone, su tamaño es reducido (10,16 x 5,33 cm), por lo que el dispositivo final será compacto.

En la siguiente figura se detallan los pines de la placa:

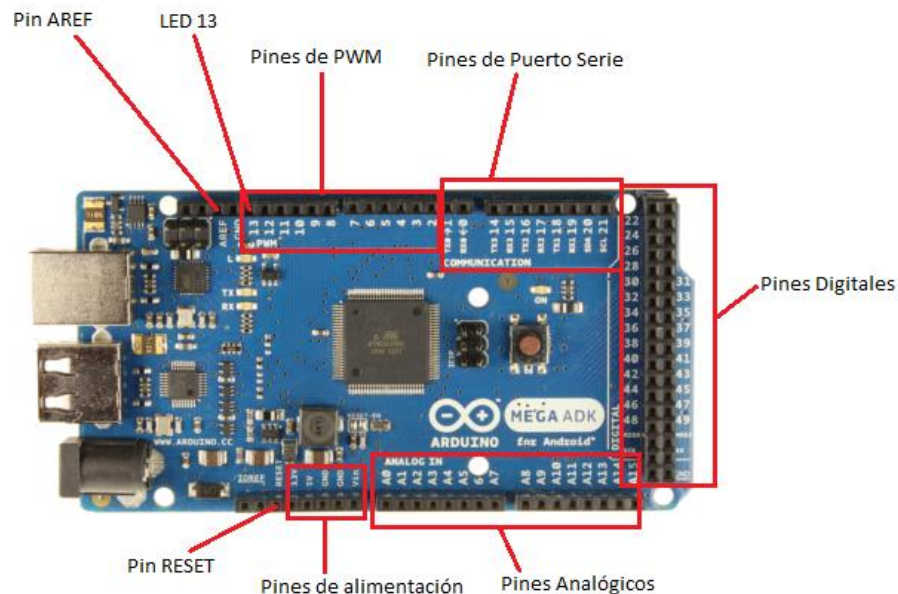


Ilustración 6: Disposición de pines

- **Pines de alimentación:**

- **VIN.** Es la entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación. Se puede proporcionar voltaje a través de este pin o, si se está alimentado a través de la conexión de 2.1mm, acceder a ella a través de este pin.
- **5V.** Es la fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V. Hay un total de tres pines de 5V, uno en la zona de alimentación y otros dos en la zona de pines digitales.
- **3.3V.** Es una fuente de voltaje a 3,3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada es de 50mA.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

- GND. Pines de toma de tierra. Tiene un total de 5: dos en la zona de alimentación, dos en la zona de los pines digitales y otro más junto a los pines de PWM.
- **Pines Digitales:**

Cada uno de los 54 pines digitales pueden utilizarse como entradas o salidas usando las funciones *pinMode()*, *digitalWrite()* y *digitalRead()*. Operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto) de 20-50kΩ. En el mismo bloque de pines, podemos encontrar arriba dos pines de 5V, y abajo otros dos pines de GND.
- **Pines de Puerto Serie:**

Los pines del conexionado serie son: Serie 0: 0 (RX0) y 1 (TX0); Serie 1: 19 (RX1) y 18 (TX1); Serie 2: 17 (RX2) y 16 (TX2); Serie 3: 15 (RX3) y 14 (TX3). Usado para recibir (RX) y transmitir (TX) datos a través de puerto serie TTL. Los pines Serie 0 (RX0) y 1 (TX0) están conectados a los pines correspondientes del chip FTDI USB-a-TTL, es decir, que están conectados a la conexión a través de USB con el ordenador.
- **Pines Analógicos:**

El Mega ADK tiene 16 entradas analógicas y cada una de ellas proporciona una resolución de 10bits (1024 valores). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función *analogReference()*.
- **Pines de PWM:**

Estos pines proporcionan una salida PWM (Pulse Wave Modulation, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) a través de la función *analogWrite()*.
- **Otros pines de la placa:**
 - AREF. Voltaje de referencia para las entradas analógicas. Usado por *analogReference()*.

- Reset. Suministra un valor de 0V para reiniciar el microcontrolador. Típicamente es usado para añadir un botón de reseteo a los shields que no dejan acceso a este botón en la placa.
- LED 13. Hay un led integrado en la placa conectado al pin digital 13. Cuando este pin tiene un valor HIGH (5V) el led se enciende y cuando este tiene un valor LOW (0V) este se apaga. Se encuentra junto a los leds de comunicación serie, que se encienden dependiendo de si transmite (TX) o recibe (RX) por cualquiera de los puertos serie.

3.1.1 ENTORNO DE DESARROLLO ARDUINO

El entorno de desarrollo para Arduino se encuentra disponible de forma gratuita en la página web de Arduino. Se puede elegir la versión dependiendo del sistema operativo utilizado. Las versiones se van actualizando temporalmente y el sistema avisa al abrir el entorno de desarrollo en caso de existir una versión actualizada.

El entorno (ver Ilustración 7) está constituido por un editor de texto en el que se escribe el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas estableciendo comunicación serie a través del canal 0 de la tarjeta.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

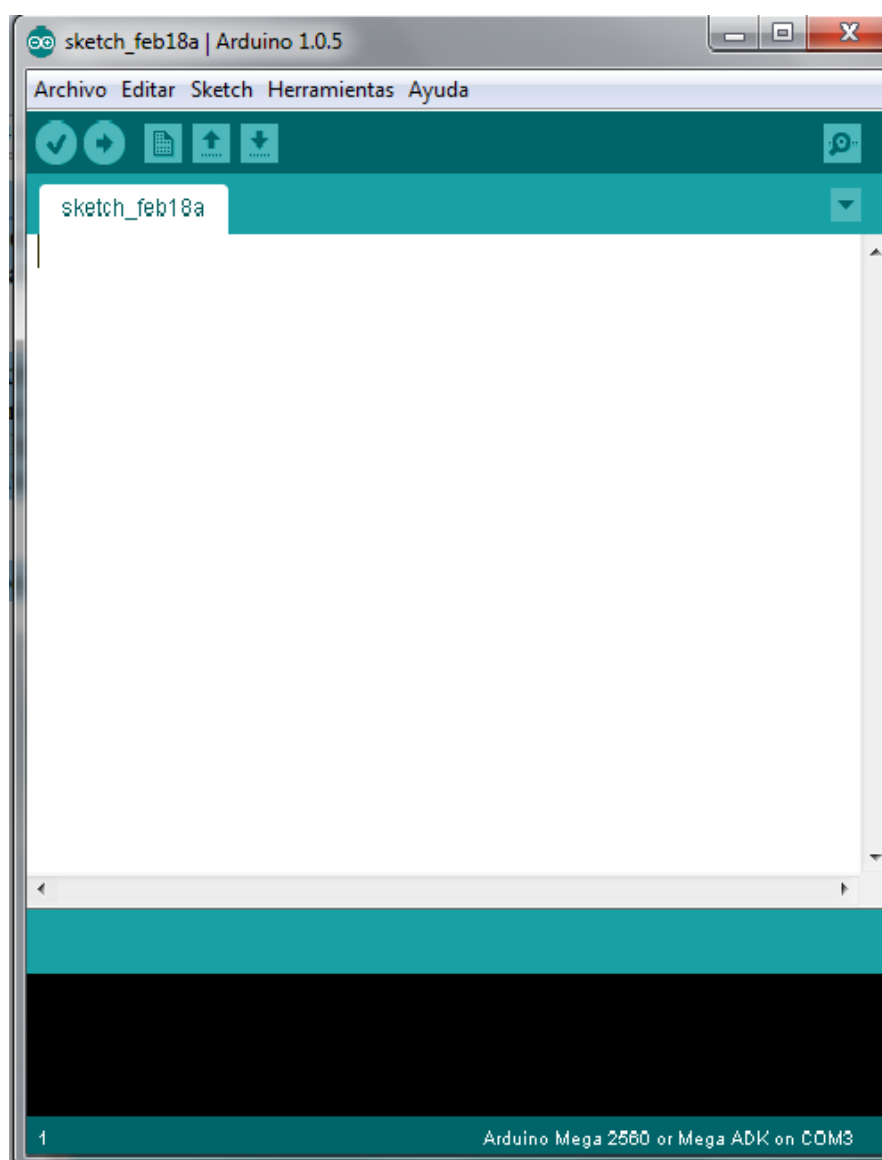


Ilustración 7: Entorno de desarrollo Arduino

Para conectar la placa con el entorno de desarrollo, se necesita seleccionar el tipo de placa, como se ve en la Ilustración 8.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

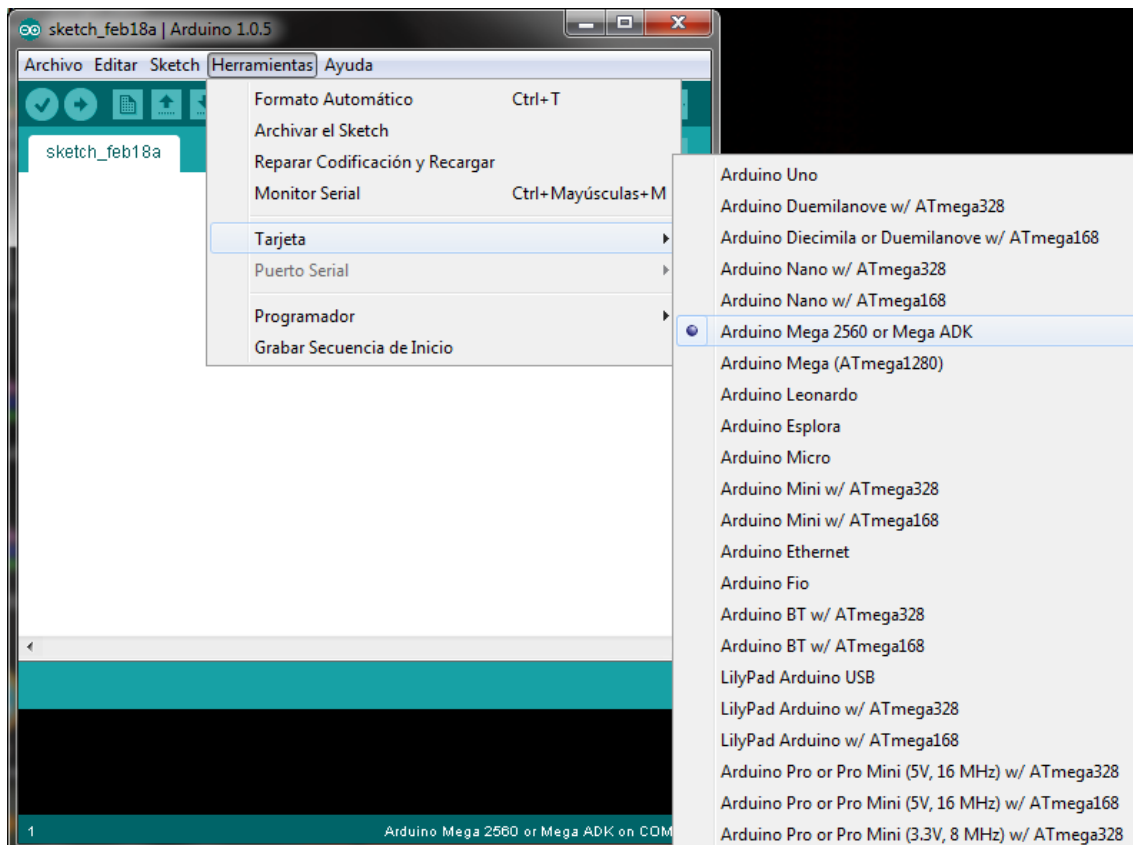


Ilustración 8: Selección del tipo de placa

Una vez seleccionada la placa es necesario seleccionar el puerto serie en el que se encuentra, para poder comenzar la comunicación. En la Ilustración 9 se muestra la selección del puerto serie.

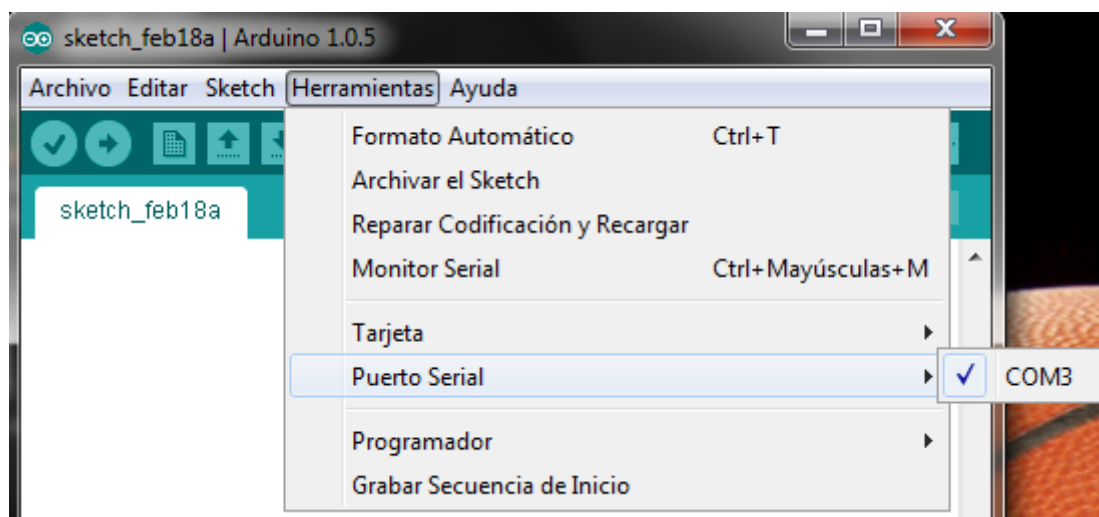


Ilustración 9: Selección del puerto serie

Arduino utiliza para escribir el software lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra los errores ocurridos al compilar o cargar el programa o si estos procesos se han realizado satisfactoriamente. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas (ver Ilustración 10) permite verificar el proceso de carga (compilar), pararlo, creación, apertura y guardado de programas, descargar el programa en la placa y la monitorización serie.

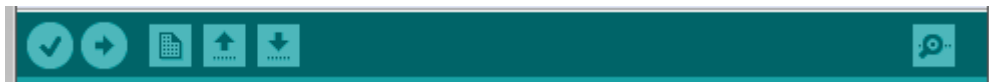


Ilustración 10: Barra de herramientas

3.1.2 LIBRERÍAS DE ARDUINO

El entorno de programación de Arduino permite la utilización de librerías, lo cual agiliza la creación de código y la simplificación del mismo. Para usarlas es necesario importarlas (ver Ilustración 11) e incluirlas al principio del código. Algunas de esas librerías se encuentran disponibles para su descarga en la página web oficial de Arduino, además de todas las instrucciones necesarias para su utilización.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

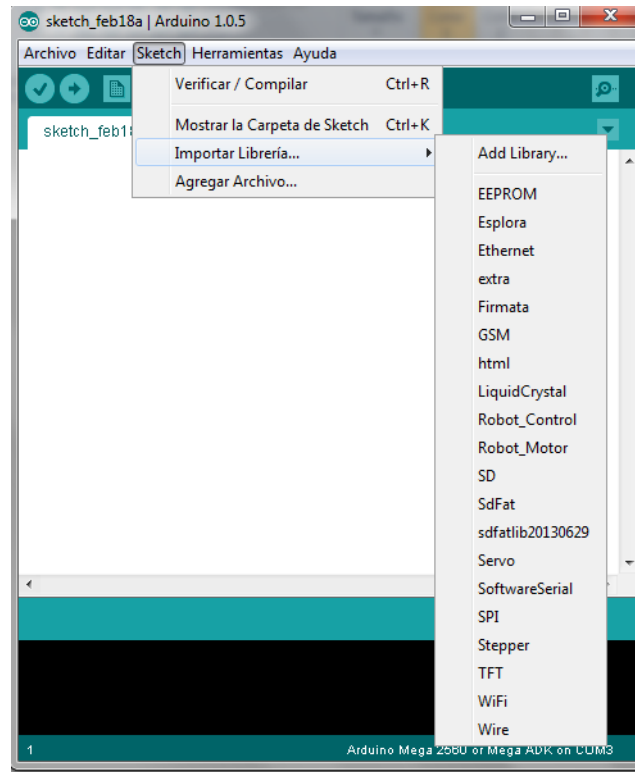


Ilustración 11: Importar librerías

3.1.3 MONITORIZACIÓN SERIE

Arduino posee su propia monitorización serie a través de la cual escribir o recibir datos por puerto serie.

Muestra los datos enviados desde la placa Arduino (placa USB o serie). Para enviar datos a la placa, se teclea el texto y se pulsa el botón "send" o "enter", que no forma parte de la trama. Es necesario seleccionar la velocidad (baud rate) en el menú desplegable y que a su vez coincida con la configurada en el `Serial.begin()` dentro del programa. En este proyecto se trabajará a una velocidad de 9600 baudios.

Cualquier mensaje que se mande imprimir en el programa, usando la función `Serial.print()` o `Serial.println()`, se muestra en este monitor serie. También realiza la función contraria: cualquier comando que se escriba se manda por puerto serie esperando a ser leído.

El puerto serie por el que lee o envía es el puerto serie 0, ya que si se puede acceder a este monitor serie es que el Arduino está conectado al ordenador a través del USB, y como se ha mencionado anteriormente, esta conexión está ligada al puerto serie 0.

3.2 SENSOR DE TEMPERATURA

3.2.1 INTRODUCCIÓN

El sensor elegido para el proyecto destinado a medir la temperatura ambiente es el sensor LM35 DZ, de la marca Texas Instruments.

El LM35 es un sensor de temperatura integrado de precisión, cuya tensión de salida es linealmente proporcional a temperatura en $^{\circ}\text{C}$ (grados centígrados). El LM35, por lo tanto, aporta una importante ventaja sobre los sensores de temperatura lineal que están calibrados en grados Kelvin: no obliga a restar una gran tensión constante para obtener grados centígrados.

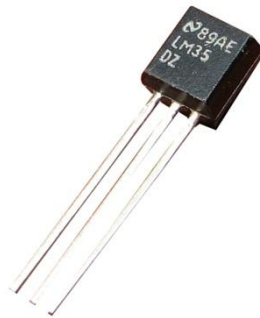


Ilustración 12: Sensor de temperatura LM35 DZ

Se fabrican distintas versiones (ver Ilustración 13), dependiendo del tipo de encapsulado o carcasa que recubre el sensor. En este caso se ha adquirido el encapsulado de plástico T0-92.

El principal motivo para su elección, aparte de su bajo coste, fue la precisión que ostenta. No requiere ninguna calibración externa o ajuste para proporcionar una precisión típica de ± 1.4 °C a temperatura ambiente y ± 3.4 °C a lo largo de su rango de temperatura (de -55 a 150 °C).

El dispositivo se ajusta y calibra durante el proceso de producción. Algunas de sus características, como la baja impedancia de salida, la precisa calibración inherente y la salida lineal, permiten la creación de circuitos de lectura o control bastante sencillos, hecho que

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

también influyó a la hora de elegirlo como sensor. El LM35 puede funcionar con alimentación simple o alimentación doble (+ y -).

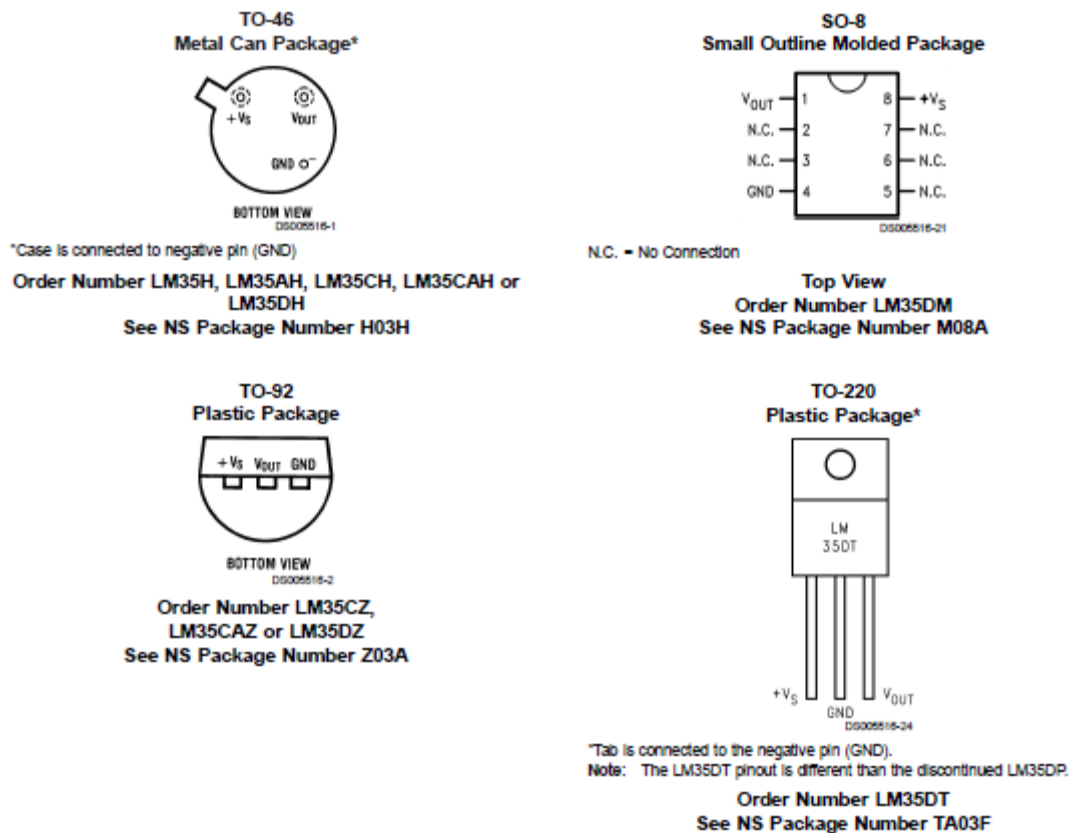


Ilustración 13: Distintos tipos de encapsulados

Requiere sólo 60 μ A para alimentarse, y bajo factor de auto-calentamiento, menos de 0,1 $^{\circ}$ C en aire estático, lo cual es de una vital importancia ya que asegura que los datos son recogidos con precisión y no se ven alterados por factores externos, en el caso de este proyecto por el calentamiento al que se verá sometido por la motocicleta.

El LM35 está preparado para trabajar en una gama de temperaturas que abarca desde los -55 $^{\circ}$ C bajo cero a 150 $^{\circ}$ C. Dependiendo de la configuración que se utilice el LM35 trabajará dentro de un rango de temperaturas u otro. La configuración más sencilla recoge temperaturas desde los +2 $^{\circ}$ C hasta los +150 $^{\circ}$ C. Como se pretende medir la temperatura ambiente será más adecuado utilizar otra configuración que permita cubrir desde los -55 $^{\circ}$ C hasta los 150 $^{\circ}$ C.

En resumen, se trata de un dispositivo de bajo coste, buena precisión, capaz de operar en un amplio rango de temperatura y con un auto-calentamiento casi inexistente, por lo que es idóneo para el proyecto.

3.2.2 PRINCIPALES CARACTERÍSTICAS

- Calibrado directamente en grados centígrados.
- Factor de escala lineal de $+10 \text{ mV} / ^\circ\text{C}$.
- $0,5^\circ\text{C}$ de precisión a $+25 ^\circ\text{C}$.
- Rango de trabajo: $-55 ^\circ\text{C}$ a $+150 ^\circ\text{C}$.
- Apropiado para aplicaciones remotas.
- Bajo coste.
- Funciona con alimentaciones entre 4V y 30V.
- Menos de $60 \mu\text{A}$ de consumo.
- Bajo auto-calentamiento ($0,08 ^\circ\text{C}$ en aire estático).
- Baja impedancia de salida, $0,1\text{W}$ para cargas de 1mA .

3.3 ACELERÓMETRO

El acelerómetro elegido para el proyecto destinado a medir las aceleraciones en las tres direcciones (ejes “x”, “y”, “z”) es el ADXL335, de la marca Sparkfun.

Este componente de baja potencia, es un acelerómetro analógico completo de 3 ejes con señales reguladas de voltaje de salida. Mide la aceleración con una escala de $\pm 3\text{G}$ y utiliza un nivel de tensión de 3.3 V . Se puede medir la aceleración de la gravedad estática en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento, choque o vibraciones.

3.3.1 ¿QUÉ ES UN ACELERÓMETRO?

Un acelerómetro es un tipo de sensor analógico transductor que detecta el movimiento o el giro, es decir, es capaz de responder con una señal eléctrica ante una perturbación inducida por la aplicación de una fuerza o la gravedad. Este dispositivo es capaz de detectar si está en horizontal o vertical o por ejemplo si los agitamos en el aire.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

El ADXL335 proporcionará a Arduino, tres voltajes proporcionales a la aceleración de cada eje “x”, “y” y “z”. Algo muy importante es que se trata de un dispositivo que se alimenta a 3.3 V, por lo que hay que tener cuidado al conectarlo a nuestro Arduino de no hacerlo a la salida de 5V, ya que dañaríamos el sensor.



Ilustración 14: Acelerómetro ADXL335

3.3.2 PRINCIPALES CARACTERÍSTICAS

- Detección de tres ejes.
- Pequeño tamaño: 4 mm x 4 mm x 1,45 mm.
- Baja energía: 350 μ A (típico).
- Una sola alimentación: 1,8 V a 3,6 V.
- 10000 g de resistencia al choque.
- Excelente estabilidad de temperatura.
- Ajuste BW con un único condensador por eje.

3.4 MEDIDORES DE ELONGACIÓN

3.4.1 ¿EN QUÉ CONSISTEN?

Se trata de dos potenciómetros que varían su valor de tensión de salida dependiendo de la elongación que tengan en cada momento.

Van colocados en la moto en la horquilla delantera y en el basculante para medir la elongación de las suspensiones. La información que se obtiene puede ser útil, por ejemplo, para saber la fuerza soportada en las frenadas y la influencia que pueda tener.



Ilustración 15: Medidor de elongación 2D

3.4.2 PRINCIPALES CARACTERÍSTICAS

No tenemos ningún tipo de datasheet de estos elementos, ya que son los correspondientes a la telemetría de la marca “2D Data Recording”, pero los hemos adaptado a la nuestra.

Sus principales características son:

- Salida lineal.
- Extensión máxima: 15 mm.
- Alimentación: 5 V.

3.5 GPS

3.5.1 INTRODUCCIÓN

El módulo receptor GPS es uno de los pilares de este proyecto. Es necesaria la utilización de un dispositivo de bajo coste con la suficiente precisión para cumplir con los objetivos fijados. El modelo elegido es el Venus GPS de la marca Sparkfun.



Ilustración 16: Venus GPS

El Sistema de Posicionamiento Global funciona a través de una red de 24 satélites en órbita sobre el planeta Tierra, con trayectorias sincronizadas para cubrir la totalidad de la superficie. El método utilizado para la localización de los módulos es el de la triangulación, es decir, cuando se desea determinar la posición, el receptor localiza automáticamente tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base a estas señales el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al módulo y de tal modo mide las distancias a los satélites. Conocidas las distancias y la posición relativa de cada satélite por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición.

De esta manera se pueden calcular las localizaciones con una gran precisión. En el caso del Venus GPS la precisión es de unos 2.5 metros, por lo que los resultados obtenidos en este proyecto son muy fiables.

Este dispositivo de posicionamiento tiene un gran rendimiento gracias al chip que lleva instalado, el Venus638FLPx, cuyas principales características son su bajo consumo, su alta sensibilidad y su rápida capacidad de reacción. Uno de los aspectos que hace a este chip tan eficiente es su motor de búsqueda, capaz de rastrear hasta las señales más débiles y de posicionarse en lugares difíciles.

Para asegurar una mayor precisión a la hora de tomar datos, se le instaló una antena externa al dispositivo, fuera de la carcasa del módulo del Arduino y la pantalla. Esto fue posible gracias a la conexión SMA del GPS que permite la instalación de este tipo de antenas. También ofrece la posibilidad de conectarse baterías externas para tener un funcionamiento autónomo, pero en este caso no fue necesario ya que el Arduino es el encargado de aplicar la energía a todos los elementos.

En resumen, el Venus GPS es un elemento muy fiable, de gran precisión y de bajo coste; condiciones que se tuvieron en cuenta a la hora de la elección de todos los componentes del proyecto.

3.5.2 PRINCIPALES CARACTERÍSTICAS

- 51 canales de detección y 14 de rastreo.
- Velocidad de actualización de 20 Hz.
- Sensibilidad de arranque en frío de -128 dBm.
- Sensibilidad de rastreo de -165 dBm.
- Arranque en frío de 29 segundos.
- Arranque en caliente de 1 segundo.
- Precisión de 2.5 metros.
- Detección y supresión de multipath (imprecisiones a la hora de calcular la posición).
- Detección y disminución de interferencias.
- Trabaja directamente con antenas pasivas y activas.
- Memoria flash interna para posible registro de datos.
- Soporta SBAS (WAAS, EGNOS, MSAS).
- Alimentación de 2.7-3.3 V.
- Dimensiones 3.8x1.8 cm.

3.5.3 FUNCIONAMIENTO

La placa de Arduino es la encargada de seleccionar qué datos de los enviados por el GPS le son útiles. Para ello, en el código de programación, se indica cuáles son las tramas con las que se va a operar.

Las tramas son las cadenas de datos que envían los módulos GPS indicando una serie de parámetros. Un ejemplo de trama sería el siguiente:

*\$GPGGA,163308.000,4019.5475,N,00347.4121,W,1,9,.086,686.3,M,51.7M,,*46*

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

Las tramas se encuentran incluidas en una serie de rangos de trabajo, categorizados por la asociación estadounidense NMEA (National Marine Electronics Association) en el protocolo 0183, por el cual los receptores GPS se comunican entre sí.

Así pues, consultando el manual de referencia de la NMEA se pueden conocer esos diferentes rangos de trabajo.

Existen nueve tipos distintos, que aportan diferentes parámetros:

- GGA: hora, posición y fijación de tipo de datos.
- GLL: latitud, longitud y tiempo de fijación de posición y estado.
- GSA: modo de operación de recepción de datos del GPS, satélites usados y valores de dilución de la posición.
- GSV: número de satélites usados con su número de identificación, elevación y ángulo de azimut.
- MSS: relación señal-ruido, fuerza de la señal, frecuencia, velocidad de bits del radio receptor.
- RMC: hora, fecha, posición, curso y velocidad.
- VTG: curso y velocidad relativas al suelo.
- ZDA: tiempo de mensajes PPS.
- 150: envío de mensaje OK.

Como los parámetros que interesaban para la realización del proyecto eran la fecha, hora, latitud, longitud y velocidad sobre el suelo, los rangos utilizados fueron el RMC y el VTG. A continuación se especifica cuáles son los datos aportados por estos rangos y su formato según el manual de la NMEA.

Por lo tanto, el Arduino señala al GPS que envíe las tramas contenidas en estos dos rangos de trabajo, y los parámetros citados anteriormente se extraen de estas tramas en los formatos que se indican en las siguientes tablas.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC Time	161229.487		hhmmss.sss
Status ¹	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation ²		degrees	E=east or W=west
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*10		
<CR> <LF>			End of message termination

Ilustración 17: Tramas RMC

Name	Example	Units	Description
Message ID	\$GPVTG		VTG protocol header
Course	309.62	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic ¹
Speed	0.13	knots	Measured horizontal speed
Units	N		Knots
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*23		
<CR> <LF>			End of message termination

Ilustración 18: Tramas VTG

3.6 SD CARD SHIELD

3.6.1 INTRODUCCIÓN

Para guardar los datos obtenidos por los distintos sensores usamos un módulo para tarjetas SD compatible con Arduino. El modelo elegido es el SD Card Shield v3.1 de la marca Seed Studio.

Este shield agrega almacenamiento a Arduino. Es compatible con tarjetas SD, SDHC o MicroSD. Utiliza un conmutador para seleccionar el tipo de tarjeta SD que hayamos introducido.

El SD Card Shield v3.1 sólo utiliza los puertos SPI de Arduino. Es apilable en algunas versiones de Arduino, así shields adicionales pueden ir en la parte superior. Hay enchufes compatibles Grove para los puertos I2C y UART a bordo.

El pin 10 se utiliza para la selección de chip, por lo tanto, es necesario asegurarse de que otros shields apilados no hagan uso de ese mismo pin.

Como ya dije antes, el SD Card Shield v3.1 es apilable para algunas versiones de Arduino, pero no lo es para la nuestra (Mega ADK), por ello, hay que hacer algunas conexiones para que el shield funcione correctamente con nuestro Arduino. Estas conexiones son las siguientes:

- Pin 10 de SD Card Shield con pin SS de SPI (pin 53) de Arduino Mega ADK.
- Pin 11 de SD Card Shield con pin MOSI de SPI (pin 51) de Arduino Mega ADK.
- Pin 12 de SD Card Shield con pin MISO de SPI (pin 50) de Arduino Mega ADK.
- Pin 13 de SD Card Shield con pin SCK de SPI (pin 52) de Arduino Mega ADK.
- Alimentación a 5 V.
- GND de SD Card Shield con GND de Arduino Mega ADK.

Esta disposición funciona bien, siempre y cuando, nos aseguremos de no asignar cualquier otra función a esos pines.

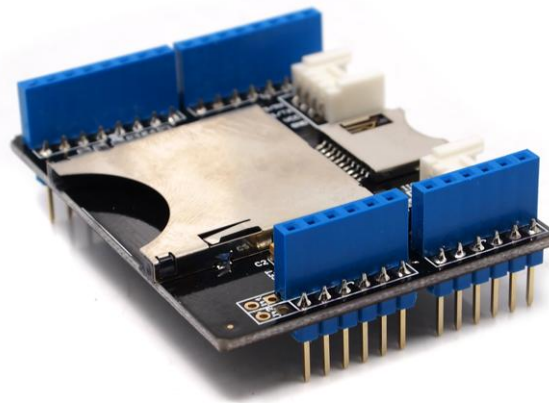


Ilustración 19: SD Card Shield

La disposición de los distintos pines en el shield es la que se muestra en la Ilustración 20:

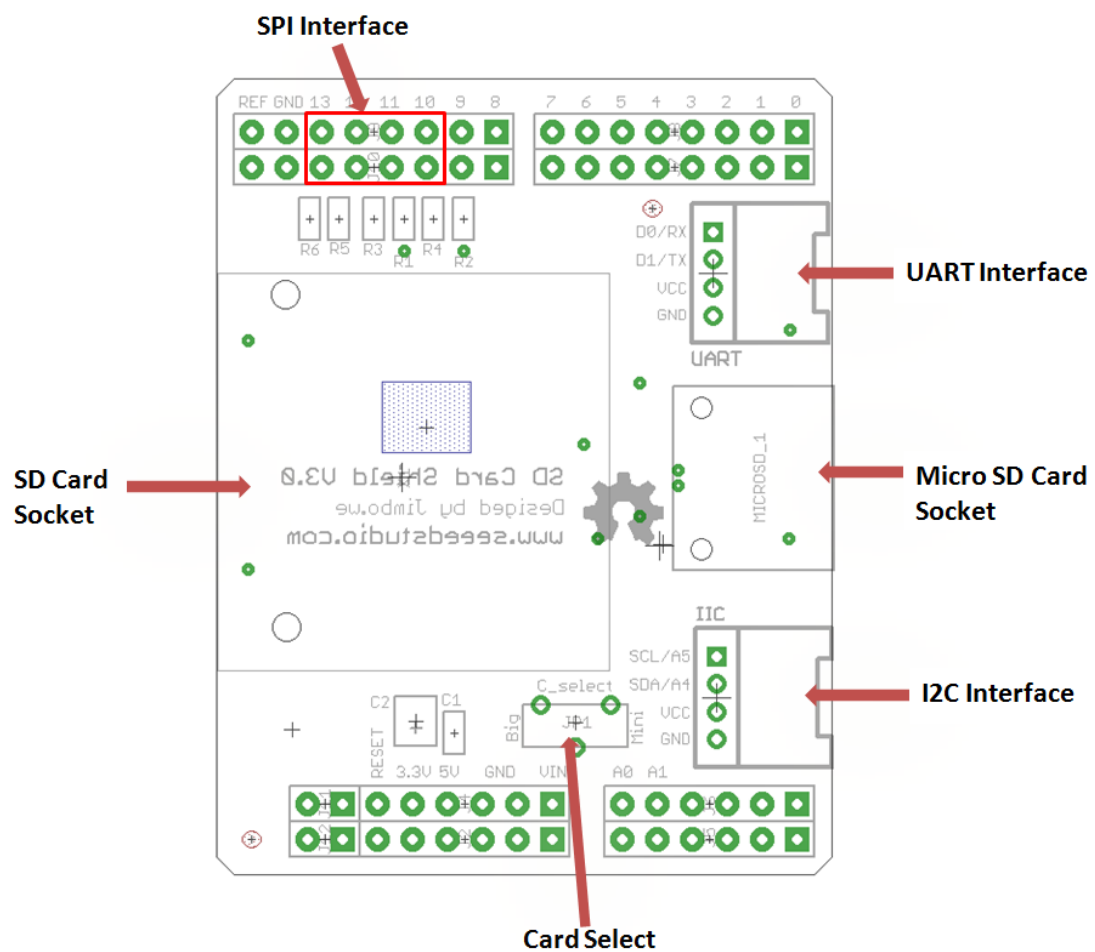


Ilustración 20: Disposición de pines

3.6.2 PRINCIPALES CARACTERÍSTICAS

- Compatible con Arduino/Seeeduino.
- Soporta tarjetas SD (≤ 2 GB), Micro SD (≤ 2 GB) y SDHC (≤ 16 GB).
- Puede ser alimentado a 3,3 V o a 5,5 V.
- Fuente de alimentación DC de 2,6 V a 3,6 V.
- Dimensiones: 57,15 mm x 44,70 mm x 19,00 mm.
- Peso: 16,6 g.

3.6.3 FUNCIONAMIENTO

Lo primero que hubo que hacer fue conectar el shield a Arduino y éste al ordenador y, con la tarjeta SD insertada en el shield, descomprimir la misma.

Una vez hecho esto, descargamos la última versión de SdFat para Arduino, que incluye distintas librerías y sketches. Para que Arduino reconozca el almacenamiento integrado de la tarjeta SD es necesario subir el sketch *SdInfo*.

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I

Álvaro Blasco Pérez

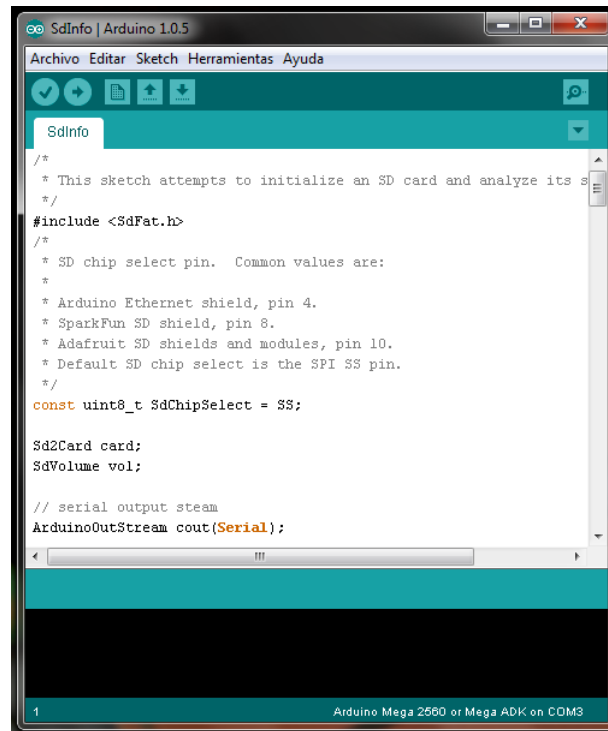


Ilustración 21: Sketch SsInfo

La forma de guardar los datos en la tarjeta SD para que luego puedan ser leídos y representados es creando un archivo de texto llamado *DATALOG.txt* que tendrá la forma del siguiente ejemplo:

21.19	2.05	1.53	53.13	105917.914	40.3356	-3.7630	5.00	282.00	43.11	64.74	1
21.14	2.56	1.53	59.04	105918.914	40.3356	-3.7630	9.26	282.80	44.68	52.03	2
1	2	3						4	5		

Ilustración 22: Ejemplo DATALOG

Donde:

1. Temperatura ambiente, pin A0.
2. Acelerómetro (x, y, z), pines A1, A2 y A3 respectivamente.
3. GPS (hora, latitud, longitud, velocidad, altitud), pin digital 10.
4. Suspensiones delantera y trasera (en mm), pines A4 y A5 respectivamente.
5. Contador del número de muestras.

Capítulo 4:

VISUALIZACIÓN DE DATOS

4. VISUALIZACIÓN DE DATOS

Como ya he venido explicando con anterioridad, este proyecto consta de dos partes: la captación de datos de los sensores y su escritura en la tarjeta SD y la lectura de los datos de la SD para su posterior tratamiento y visualización.

De ésta segunda parte se ha encargado otra persona con la que ha sido necesario trabajar mano a mano, ya que, tanto yo he tenido que saber en todo momento qué tipo de información necesita y la forma en que debía ser escrita para su correcta lectura como él debía saber las posibilidades que se ofrecían de lectura y el modo en el que debía tratar la información que llegaba en la tarjeta SD.

Ésta segunda parte estaba compuesta por un Arduino Mega ADK idéntico al de captación de datos y otro SD Card Shield para leer la tarjeta SD que viene de la captación de datos.

El trabajo que se ha llevado a cabo en la parte de visualización de datos ha sido leer los datos del archivo de texto *DATALOG.txt* , hacer una interfaz gráfica legible y tratar esa información para sacar las conclusiones correctas de cualquier recorrido en pista.

Capítulo 5:

SIMULACIÓN REAL

5. SIMULACIÓN REAL

Durante las distintas pruebas que hemos llevado a cabo, hemos realizado recorridos en coche por la calle. Los resultados obtenidos nos han hecho realizar distintos cambios para que todos los sensores funcionasen a la vez de manera correcta.

Como ya expliqué anteriormente, los datos de los sensores se guardan en un archivo de texto de la siguiente forma:

21.19	2.05	1.53	53.13	105917.914	40.3356	-3.7630	5.00	282.00	43.11	64.74	1
21.14	2.56	1.53	59.04	105918.914	40.3356	-3.7630	9.26	282.80	44.68	52.03	2
1	2	3					4	5			

Ilustración 23: Ejemplo DATALOG

Donde:

1. Temperatura ambiente.
2. Acelerómetro (x, y, z).
3. GPS (hora, latitud, longitud, velocidad, altitud).
4. Suspensiones delantera y trasera (en mm).
5. Contador del número de muestras.

Ésta es la información que puedo visualizar de la parte de captación de datos tras realizar una simulación. Con esta información se realizan todo tipo de gráficas con las que se puede ver la última ruta, compararla con otras guardadas, visualizar el recorrido en Google Maps, etc.

En la Ilustración 24 se puede observar un recorrido que hicimos en la zona cercana al Campus de la Universidad Carlos III en Leganés. Los distintos colores en el recorrido corresponden a la velocidad con la que circulábamos.



Ilustración 24: Recorrido en simulación real

Capítulo 6:

CONCLUSIONES

6. CONCLUSIONES

El proyecto ha sido muy gratificante, ya que partía de no tener ninguna base de conocimientos de Arduino y al final he podido realizar una telemetría usando únicamente esta tecnología. Es ahí donde veo la dificultad y, por lo tanto, la parte más importante de ésta trabajo.

Debido a la flexibilidad que ofrecen las placas Arduino, resulta muy interesante el uso de las mismas para este tipo de trabajos, ya que, podemos ampliar el trabajo realizado sin necesidad de realizar grandes cambios en la estructura o el código. Del mismo modo que es adaptable con otros tipos de programas o entornos con los que podemos trabajar en paralelo.

Lo más interesante del proyecto es que puede y, en mi opinión, debe ser ampliado y mejorado después de hacer simulaciones reales con el sistema ya montado en una telametría para competir. Debido a que no hemos podido probarlo montado en una motocicleta quizás no se hayan podido sacar tantas conclusiones como sería deseable.

En cualquier caso, el proyecto es un inicio de lo que podría llegar a ser una buena telemetría y los resultados obtenidos en nuestras simulaciones han sido correctos.

Durante la realización de este proyecto pudimos comparar telemetrías ya existentes como, por ejemplo, la de 2D Data Recording que tienen un alto coste y realizan las mismas funciones. En mi opinión, merece mucho más la pena usar la telemetría que hemos realizado, ya que el coste de una placa Arduino es asequible para cualquier bolsillo a diferencia de lo que el mercado ofrece y, además, nos ofrece un amplio abanico de mejoras que con otros sistemas no se pueden realizar.

Capítulo 7:

FUTUROS TRABAJOS

7. FUTUROS TRABAJOS

Como futuros trabajos o ampliaciones posibles a este proyecto propondría las siguientes:

- Realización de una caja rígida donde iría el Arduino, los shields que se usen, el cableado, la alimentación y un interruptor, con un fácil acceso para insertar y extraer la tarjeta SD; llevada a cabo con una impresora 3D.
- Montar el sistema en una moto y realizar pruebas con ella.
- Montaje de una sonda de líquido para medir la temperatura del agua de refrigeración.
- Montaje de un tacómetro.
- Añadir un módulo de radiofrecuencia con una antena emisora en la moto y otra receptora en el box para poder visualizar los datos en tiempo real.

Capítulo 8:

BIBLIOGRAFÍA

8. BIBLIOGRAFÍA

- Página web oficial de arduino
<http://www.arduino.cc/>
- Librería serie:
<http://arduino.cc/es/Reference/serial>
- SD Card Shield:
http://www.seeedstudio.com/wiki/SD_Card_Shield_V3.0
- Página web oficial de Sparkfun:
<https://www.sparkfun.com/>
- Página web oficial de Texas Instruments:
<http://www.ti.com/>
- Acelerómetro Sparkfun:
<https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>
- Página web oficial de 2D Data Recording:
<http://2d-datarecording.com/>

Capítulo 9:

ANEXOS

9. ANEXOS

9.1 ANEXO I: CÓDIGO

```
#include <SD.h>
```

```
#include <SoftwareSerial.h>
```

```
#include <string.h>
```

```
#include <stdlib.h>
```

```
SoftwareSerial gpsSerial(10, 11); // RX, TX (TX not used)
```

```
const int sentenceSize = 80;
```

```
File dataFile;
```

```
int led = 13;
```

```
float tempC;
```

```
int tempPin = 0; // Definimos la entrada en pin A0
```

```
const int xPin = 1; //Pines analogicos de lectura
```

```
const int yPin = 2;
```

```
const int zPin = 3;
```

```
int minVal = 265; // Valores mínimos y máximos del acelerometro en reposo
```

```
int maxVal = 402;
```

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

double x; // para guardar los valores calculados

double y;

double z;

float suspDel;

int suspDelPin = 4; // Definimos la entrada en pin A4

float suspTra;

int suspTraPin = 5; // Definimos la entrada en pin A5

char sentence[sentenceSize];

char N;

char W;

char h;

float latit;

float longt;

double v;

double alt;

int cont=0;

void setup()

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

```
{

  Serial.begin(9600);

  gpsSerial.begin(9600);

  pinMode(led, OUTPUT);

  Serial.print("Iniciando SD card...");

  pinMode(53, OUTPUT);

  pinMode(led, OUTPUT);

  digitalWrite(53, HIGH);


  if (!SD.begin(53))

  {

    Serial.println("Fallo comunicacion o no existe SD");

    digitalWrite(led, HIGH);

    return;

  }


  digitalWrite(led, LOW);

  Serial.println("SD iniciada.");


  dataFile = SD.open("datalog.txt", FILE_WRITE);

}


void loop()
```

```
{  
  
    // Lee el valor desde el sensor  
  
    tempC = analogRead(tempPin);  
  
  
    // Convierte el valor a temperatura  
  
    tempC = (5.0 * tempC * 100.0)/1024.0;  
  
  
    // Envia el dato al puerto serial  
  
  
  
    int xRead = analogRead(xPin); //Lee los valores analogicos del acelerometro  
  
    int yRead = analogRead(yPin);  
  
    int zRead = analogRead(zPin);  
  
  
  
    // mapea los valores leidos a un rango -90 a 90 grados (- $\pi$  a  $\pi$ )  
  
    int xAng = map(xRead, minVal, maxVal, -90, 90);  
  
    int yAng = map(yRead, minVal, maxVal, -90, 90);  
  
    int zAng = map(zRead, minVal, maxVal, -90, 90);  
  
  
    //Convertimos los radianes a grados  
  
    x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI);  
  
    y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);  
  
    z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);  
}
```

```
// Lee el valor desde el sensor

suspDel = analogRead(suspDelPin);

suspDel = (75.0/1023.0)*suspDel;


// Lee el valor desde el sensor

suspTra = analogRead(suspTraPin);

suspTra = (75.0/1023.0)*suspTra;


static int i = 0;

if (gpsSerial.available())

{

    char ch = gpsSerial.read();

    if (ch != '\n' && i < sentenceSize)

    {

        sentence[i] = ch;

        i++;

    }

    else

    {

        sentence[i] = '\0';

        i = 0;

        displayGPS();

    }

}
```

```
}
```

```
cont++;
```

```
dataFile = SD.open("datalog.txt", FILE_WRITE);
```

```
if (dataFile)
```

```
{
```

```
    digitalWrite(led, LOW);
```

```
    dataFile.print(tempC);
```

```
    dataFile.print(" ");
```

```
    dataFile.print(x);
```

```
    dataFile.print(" ");
```

```
    dataFile.print(y);
```

```
    dataFile.print(" ");
```

```
    dataFile.print(z);
```

```
    dataFile.print(" ");
```

```
    dataFile.print(suspDel);
```

```
    dataFile.print(" ");
```

```
    dataFile.print(suspTra);
```

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

```
dataFile.print(" ");
```

```
displayGPS();
```

```
dataFile.println(cont);
```

```
delay(500);
```

```
dataFile.close();
```

```
Serial.print(tempC);
```

```
Serial.print(" ");
```

```
Serial.print(x);
```

```
Serial.print(" ");
```

```
Serial.print(y);
```

```
Serial.print(" ");
```

```
Serial.print(z);
```

```
Serial.print(" ");
```

```
Serial.print(suspDel);
```

```
Serial.print(" ");
```

```
Serial.print(suspTra);
```

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

```
Serial.print(" ");

Serial.println(cont);

}

else

{

    Serial.println("Fallo comunicacion.txt");

    Serial.println("Revise conexion");

    digitalWrite(led, HIGH);

    delay(500);

}

}

void displayGPS()

{

    char field[20];

    getField(field, 0);

    if (strcmp(field, "$GPRMC") == 0)

    {
```


DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

```
//HORA Serial.print("Hora: ");

getField(field, 1); // hhmmss.sss

h=atof((const char*)field);

dataFile.print(h,3);

dataFile.print(" ");

Serial.print(h,3);

Serial.print(" ");


//LATITUD Serial.print("Lat: ");

getField(field, 3); // number

double lat = atof((const char*)field);//Convierte un array de char a un float

float m=lat*10000;//Latitud con todos sus digitos

lat =(lat/100);//Obtenemos los dos primeros digitos

int grad = (int)lat;//Esos dos dígitos son mis grados

m = m-(grad*1000000);//Obtenemos minutos

m = m/600000;//Obtenemos la conversion de minutos a grados

latit = grad+m;//sumamos grados mas minutos convertidos a grados


//Damos a la latitud su signo según N/S

getField(field, 4); //N/S

if((field,4)==N)

{

    latit=+latit;
```


DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

```
{  
  
    longt=-longt;  
  
}  
  
else  
  
{  
  
    longt=+longt;  
  
}  
  
dataFile.print(longt,4);  
  
dataFile.print(" ");  
  
Serial.print(longt,4); //( ,4) sirve para imprimir el numero de decimales que uno necesite  
  
Serial.print(" ");  
  
//getField(field, 6); // E/W  
  
//Serial.print(field);  
  
//Serial.print(" ");  
  
//VELOCIDAD EN MILLAS NAUTICAS  
  
getField(field, 7); // number velocidad en knots  
  
v= atof((const char*)field);  
  
v=v*1.852000002; //VELOCIDAD EN KM/H  
  
dataFile.print(v,2);  
  
dataFile.print(" ");  
  
Serial.print(v,2);  
  
Serial.print(" ");  
  
}
```

```
getField(field, 0);

if (strcmp(field, "$GPGLL") == 0)

{

    //ALTITUD

    getField(field, 9);

    alt= atof((const char*)field);

    dataFile.print(alt,2);

    dataFile.print(" ");

    Serial.print(alt,2);

    Serial.print(" ");

}

dataFile.close();

}
```

```
void getField(char* buffer, int index)

{

    int sentencePos = 0;

    int fieldPos = 0;

    int commaCount = 0;

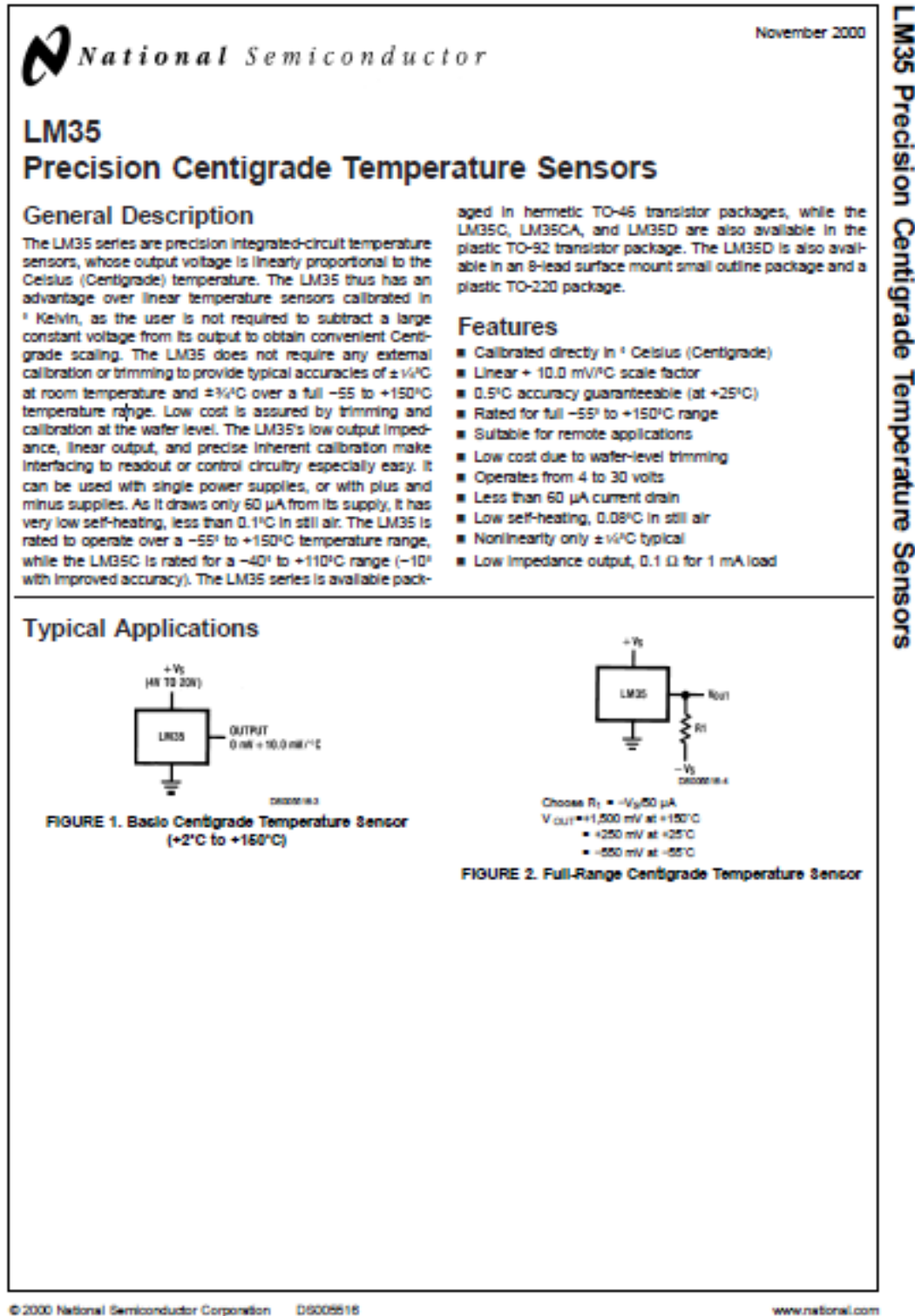
    while (sentencePos < sentenceSize)

    {
```

```
if (sentence[sentencePos] == ',')  
  
    {  
  
        commaCount++;  
  
        sentencePos++;  
  
    }  
  
if (commaCount == index)  
  
    {  
  
        buffer[fieldPos] = sentence[sentencePos];  
  
        fieldPos++;  
  
    }  
  
    sentencePos++;  
  
}  
  
buffer[fieldPos] = '\\0';  
  
}
```

9.2 ANEXO II: DATASHEETS

9.2.1 SENSOR DE TEMPERATURA LM 35



Connection Diagrams

**TO-48
Metal Can Package***



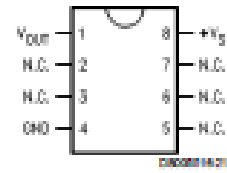
BOTTOM VIEW
DISCONTINUED

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

**SO-8
Small Outline Molded Package**



DISCONTINUED

N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

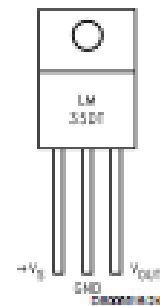
**TO-92
Plastic Package**



BOTTOM VIEW
DISCONTINUED

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

**TO-220
Plastic Package***



DISCONTINUED

*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)		TO-92 and TO-220 Package, (Soldering, 10 seconds)		260°C
If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.		SO Package (Note 12)		
Supply Voltage		Vapor Phase (60 seconds)		215°C
Output Voltage		Infrared (15 seconds)		220°C
Output Current		ESD Susceptibility (Note 11)		2500V
Storage Temp.:		Specified Operating Temperature Range: T _{MIN} to T _{MAX} (Note 2)		
TO-46 Package,		LM35, LM35A		-55°C to +150°C
TO-92 Package,		LM35C, LM35CA		-40°C to +110°C
SO-8 Package,		LM35D		0°C to +100°C
TO-220 Package,				
Lead Temp.:				
TO-46 Package, (Soldering, 10 seconds)				300°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T _A = +25°C	±0.2	±0.5		±0.2	±0.5		°C
	T _A = -10°C	±0.3			±0.3		±1.0	°C
	T _A = T _{MAX}	±0.4	±1.0		±0.4	±1.0		°C
	T _A = T _{MIN}	±0.4	±1.0		±0.4		±1.5	°C
Nonlinearity (Note 8)	T _{MIN} < T _A < T _{MAX}	±0.18		±0.36	±0.16		±0.3	°C
Sensor Gain (Average Slope)	T _{MIN} < T _A < T _{MAX}	+10.0	+9.8, +10.1		+10.0		+9.8, +10.1	mV/°C
Load Regulation (Note 3) 0 ≤ I _L ≤ 1 mA	T _A = +25°C	±0.4	±1.0		±0.4	±1.0		mV/mA
	T _{MIN} < T _A < T _{MAX}	±0.6		±3.0	±0.6		±3.0	mV/mA
Line Regulation (Note 3)	T _A = +25°C	±0.01	±0.05		±0.01	±0.05		mV/V
	4V ≤ V _{IN} ≤ 30V	±0.02		±0.1	±0.02		±0.1	mV/V
Quiescent Current (Note 9)	V _{IN} = +5V, +25°C	56	67		56	67		µA
	V _{IN} = +5V	106		131	81		114	µA
	V _{IN} = +30V, +25°C	56.2	68		56.2	68		µA
	V _{IN} = +30V	106.6		133	81.6		118	µA
Change of Quiescent Current (Note 3)	4V ≤ V _{IN} ≤ 30V, +25°C	0.2	1.0		0.2	1.0		µA
	4V ≤ V _{IN} ≤ 30V	0.6		2.0	0.6		2.0	µA
Temperature Coefficient of Quiescent Current		+0.38		+0.6	+0.38		+0.6	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, I _L = 0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T _A = T _{MAX} for 1000 hours	±0.08			±0.08			°C

DISEÑO DE LA TELEMETRÍA PARA UNA MOTOCICLETA DE COMPETICIÓN I
Álvaro Blasco Pérez

LM35

Electrical Characteristics								
(Notes 1, 5)								
Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^{\circ}\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		$^{\circ}\text{C}$
	$T_A = -10^{\circ}\text{C}$	± 0.5			± 0.5		± 1.5	$^{\circ}\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	$^{\circ}\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2.0	$^{\circ}\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^{\circ}\text{C}$				± 0.6	± 1.5		$^{\circ}\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2.0	$^{\circ}\text{C}$
	$T_A = T_{\text{MIN}}$				± 0.9		± 2.0	$^{\circ}\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.6	± 0.2		± 0.6	$^{\circ}\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	$+10.0$	$+9.8$, $+10.2$		$+10.0$		$+9.8$, $+10.2$	mV/ $^{\circ}\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^{\circ}\text{C}$	± 0.4	± 2.0		± 0.4	± 2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.6		± 6.0	± 0.6		± 6.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^{\circ}\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4 \text{ V} \leq V_{\text{IN}} \leq 30 \text{ V}$	± 0.02		± 0.2	± 0.02		± 0.2	mV/V
Quiescent Current (Note 9)	$V_{\text{IN}} = +5 \text{ V}$, $+25^{\circ}\text{C}$	56	80		56	80		μA
	$V_{\text{IN}} = +5 \text{ V}$	106		168	81		138	μA
	$V_{\text{IN}} = +30 \text{ V}$, $+25^{\circ}\text{C}$	56.2	82		56.2	82		μA
	$V_{\text{IN}} = +30 \text{ V}$	106.6		181	81.6		141	μA
Change of Quiescent Current (Note 3)	$4 \text{ V} \leq V_{\text{IN}} \leq 30 \text{ V}$, $+25^{\circ}\text{C}$	0.2	2.0		0.2	2.0		μA
	$4 \text{ V} \leq V_{\text{IN}} \leq 30 \text{ V}$	0.6		3.0	0.6		3.0	μA
Temperature Coefficient of Quiescent Current		$+0.38$		$+0.7$	$+0.38$		$+0.7$	$\mu\text{A}/^{\circ}\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^{\circ}\text{C}$
Long Term Stability	$T_A = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^{\circ}\text{C}$

Note 1: Unless otherwise noted, these specifications apply: $-55^{\circ}\text{C} \leq T_J \leq +150^{\circ}\text{C}$ for the LM35 and LM35A; $-40^{\circ}\text{C} \leq T_J \leq +110^{\circ}\text{C}$ for the LM35C and LM35CA; and $0^{\circ}\text{C} \leq T_J \leq +100^{\circ}\text{C}$ for the LM35D. $V_{\text{IN}} = +5 \text{ V}$ dc and $I_{\text{LOAD}} = 50 \mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^{\circ}\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-45 package is 400°C/W (junction to ambient, and 24°C/W junction to case). Thermal resistance of the TO-82 package is 100°C/W (junction to ambient). Thermal resistance of the small outline molded package is 220°C/W (junction to ambient). Thermal resistance of the TO-220 package is 90°C/W (junction to ambient). For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in boldface apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10 \text{ mV}/^{\circ}\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^{\circ}\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of Figure 1.

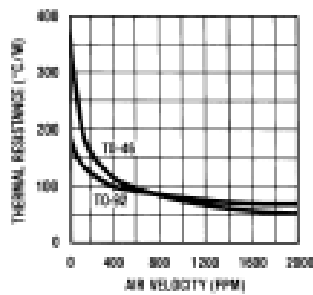
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a $1.5 \text{ k}\Omega$ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

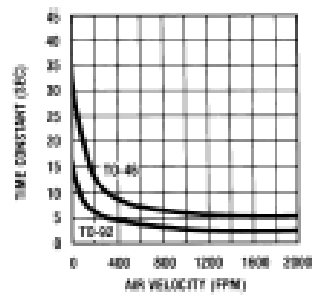
Typical Performance Characteristics

**Thermal Resistance
Junction to Air**



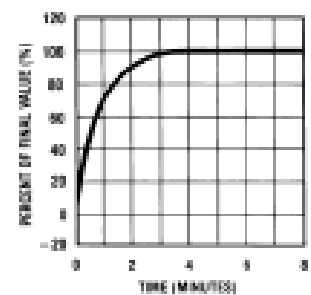
DA000016A.20

Thermal Time Constant



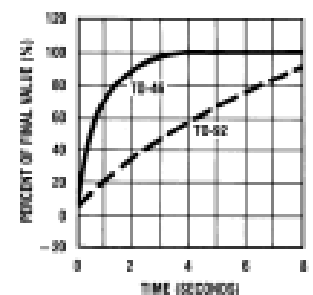
DA000016A.26

**Thermal Response
In Still Air**



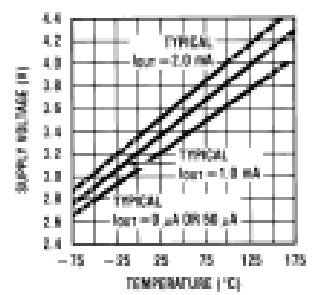
DA000016A.27

**Thermal Response In
Stirred Oil Bath**



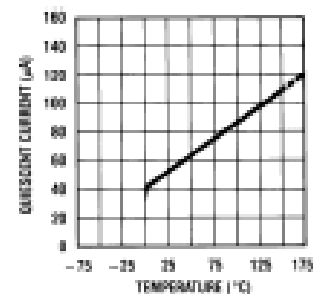
DA000016A.28

**Minimum Supply
Voltage vs. Temperature**



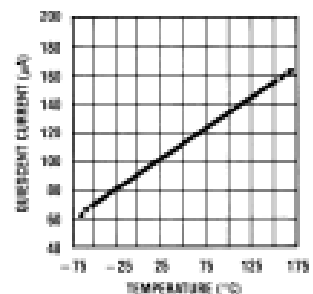
DA000016A.29

**Quiescent Current
vs. Temperature
(In Circuit of Figure 1.)**



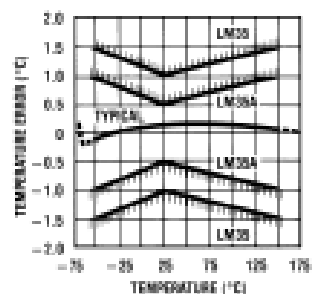
DA000016A.30

**Quiescent Current
vs. Temperature
(In Circuit of Figure 2.)**



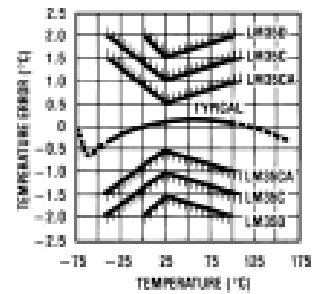
DA000016A.31

**Accuracy vs. Temperature
(Guaranteed)**



DA000016A.32

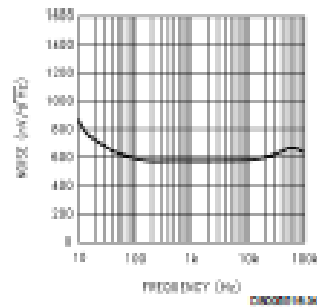
**Accuracy vs. Temperature
(Guaranteed)**



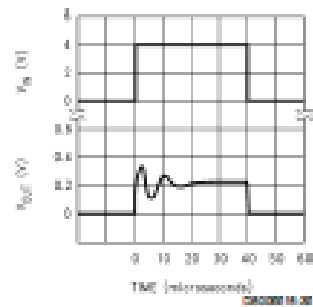
DA000016A.33

Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92**, small heat fin	SO-8 no heat sink	SO-8** small heat fin	TO-220 no heat sink
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	80°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	100°C/W	90°C/W	20°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Stirred oil	50°C/W	30°C/W	40°C/W	40°C/W			
(Clamped to metal, infinite heat sink)		(24°C/W)				(50°C/W)	

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications

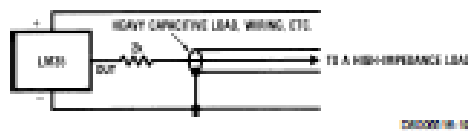


FIGURE 3. LM35 with Decoupling from Capacitive Load

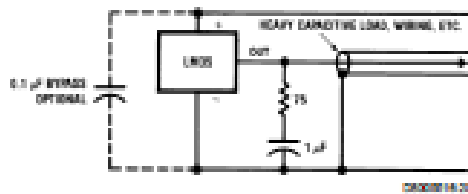


FIGURE 4. LM35 with R-C Damper

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor, see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see Figure 4.

When the LM35 is applied with a 200Ω load resistor as shown in Figure 5, Figure 6 or Figure 8 it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc., as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in Figure 13, Figure 14, and Figure 15.

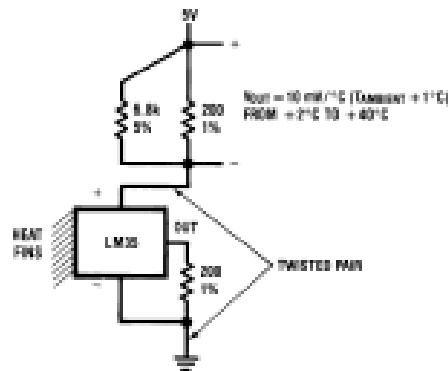


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)

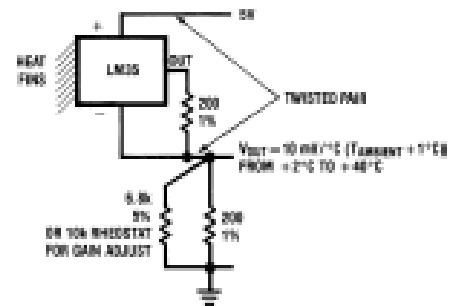


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

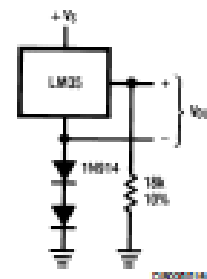


FIGURE 7. Temperature Sensor, Single Supply, -65° to +150°C

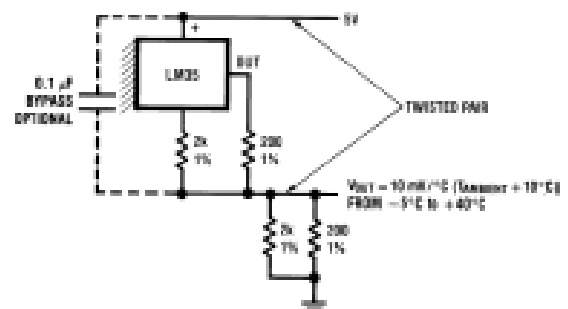


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

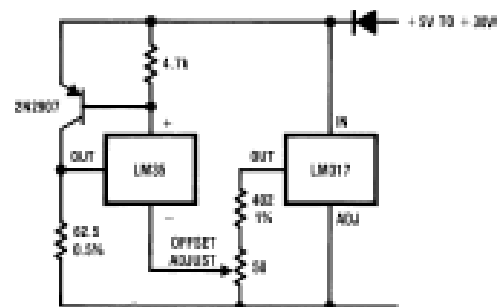
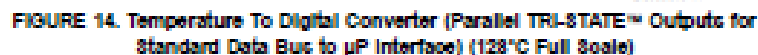
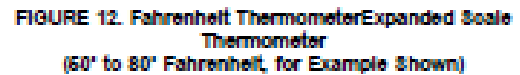


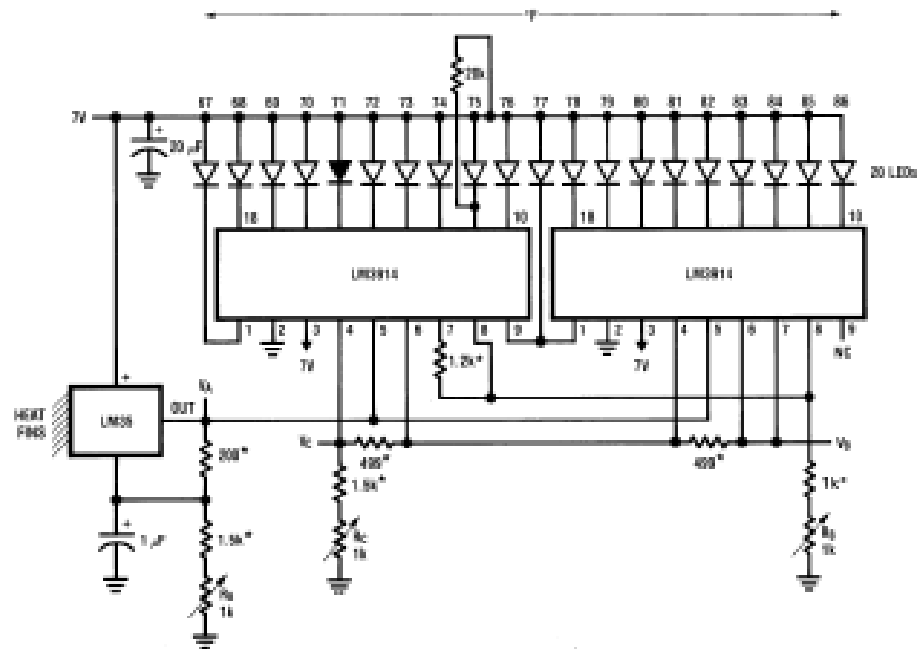
FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)



FIGURE 11. Centigrade Thermometer (Analog Meter)



Typical Applications (Continued)



*=1% or 2% film resistor
 Trim R_{10} for $V_{OUT}=0.075V$
 Trim R_{11} for $V_{OUT}=1.665V$
 Trim R_{12} for $V_{OUT}=0.075V + 100mV/°C \times T_{ambient}$
 Example, $V_{OUT}=0.275V$ at $22°C$

FIGURE 16. Bar-Graph Temperature Display (Dot Mode)

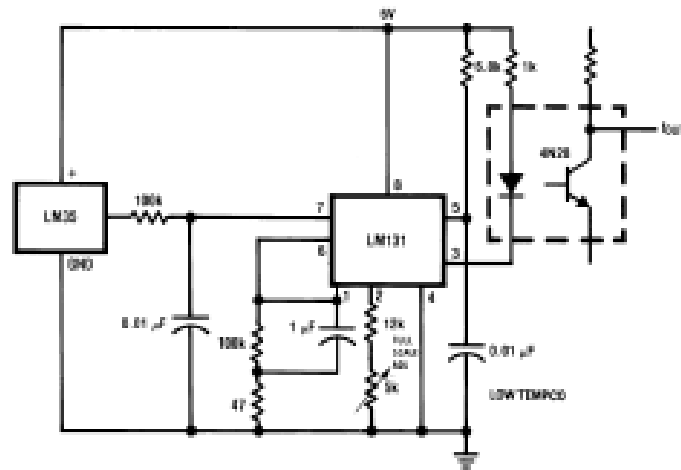
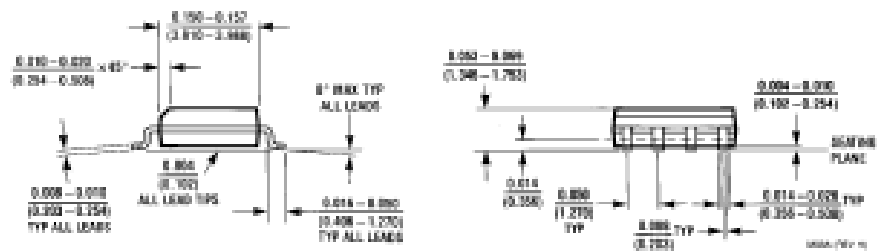
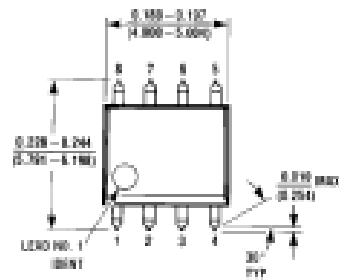
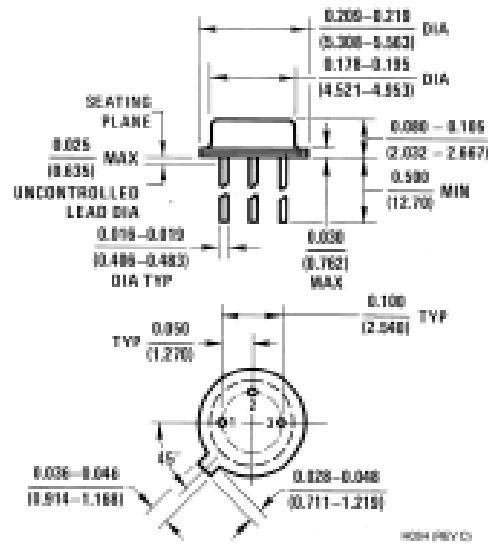


FIGURE 18. LM35 With Voltage-To-Frequency Converter And Isolated Output
 ($2°C$ to $+150°C$; 20 Hz to 1600 Hz)

Block Diagram



Physical Dimensions Inches (millimeters) unless otherwise noted



Technical drawing of a mechanical part, showing two views: a front view (top) and a side view (bottom).

Front View Dimensions:

- Overall width: 0.240-0.280 [6.10-7.12]
- Top width: 0.100-0.120 [2.54-3.05]
- Top width: 0.350-0.350 [8.89-8.89]
- Top width: 0.140-0.153 [3.78-3.89]
- Top width: 0.090-0.110 [2.29-2.79]
- Top width: 0.180-0.210 [4.65-5.33]
- Left height: 0.400 ±0.010 -0.005 +0.06 [-0.13]
- Left height: 0.100-0.120 TYP [2.54-3.05]
- Left height: 0.060-0.075 TYP [1.52-1.91]
- Left height: 0.300 ±0.015 -0.005 +0.08 [-0.13]
- Left height: 0.801-0.807 TYP [0.25-0.178]
- Center hole diameter: 0.045-0.055 TYP [1.14-1.40]
- Right hole diameter: 0.130-0.160 TYP [3.30-4.06]
- Right hole diameter: 0.040-0.055 TYP [1.32-1.40]
- Right hole diameter: 0.027-0.037 TYP [0.69-0.94]
- Overall length: 1.005-1.035 [25.53-26.29]
- Pin diameter: PIN = 1.10

Side View Dimensions:

- Top width: 0.175-0.185 [4.45-4.70]
- Top width: 0.040-0.052 [1.32-1.32]
- Top width: 0.525-0.555 [13.34-14.10]
- Top width: 0.015 ±0.007 -0.001 +0.18 [0.38 ±0.03]
- Top width: 0.105 ±0.010 -0.015 +0.35 [2.67 ±0.38]
- Angle: 7°
- Angle: 7°
- Angle: 0°-1°
- Feature: SEATING PLANE
- Feature: TAPERED SIDES 1°

TLC-MS (MWH A)

EJECTION MARK
Ø1.6 MAX
W0.38 MAX

SEATING PLANE

2.29 MAX
(UNCONTROLLED LEAD DIA)

0.55
0.40 TYP

1.27 ±0.05

2.54 ±0.1

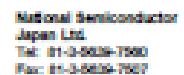
2.21
2.18

0.58
0.58

DIMENSIONS ARE IN MILLIMETERS

1100 1101 1102

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



86

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

9.2.2 ACELERÓMETRO ADXL 335



Small, Low Power, 3-Axis $\pm 3g$ Accelerometer

ADXL335

FEATURES

- 3-axis sensing
- Small, low profile package
 - 4 mm x 4 mm x 1.45 mm LFCSP
- Low power : 350 μ A (typical)
- Single-supply operation: 1.8 V to 3.6 V
- 10,000 g shock survival
- Excellent temperature stability
- BW adjustment with a single capacitor per axis
- RoHS/WEEE lead-free compliant

APPLICATIONS

- Cost sensitive, low power, motion- and tilt-sensing applications
- Mobile devices
- Gaming systems
- Disk drive protection
- Image stabilization
- Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of $\pm 3g$. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_x , C_y , and C_z capacitors at the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm x 4 mm x 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

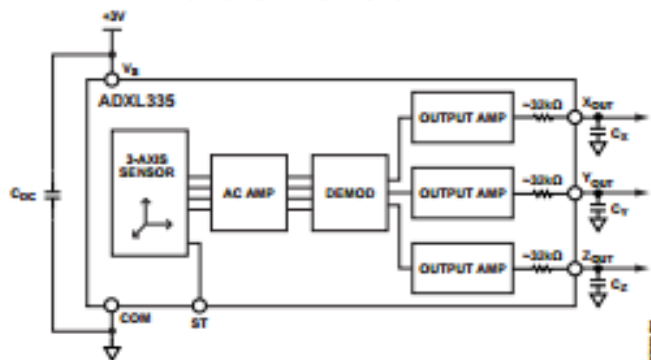


Figure 1.

Rev. 0

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2009 Analog Devices, Inc. All rights reserved.

ADXL335

TABLE OF CONTENTS

Features	1	Performance	10
Applications	1	Applications Information	11
General Description	1	Power Supply Decoupling	11
Functional Block Diagram	1	Setting the Bandwidth Using C_x , C_y , and C_z	11
Revision History	2	Self Test	11
Specifications	3	Design Trade-Offs for Selecting Filter Characteristics:	
Absolute Maximum Ratings	4	The Noise/BW Trade-Off	11
ESD Caution	4	Use with Operating Voltages Other than 3 V	11
Pin Configuration and Function Descriptions	5	Axes of Acceleration Sensitivity	12
Typical Performance Characteristics	6	Layout and Design Recommendations	13
Theory of Operation	10	Outline Dimensions	14
Mechanical Sensor	10	Ordering Guide	14

REVISION HISTORY

1/09—Revision 0: Initial Version

SPECIFICATIONS

T_A = 25°C, V_S = 3 V, C_X = C_Y = C_Z = 0.1 µF, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		g
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ¹			±1		%
SENSITIVITY (RATIOMETRIC)²	Each axis				
Sensitivity at X _{out} , Y _{out} , Z _{out}	V _S = 3 V	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	V _S = 3 V		±0.01		%/°C
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X _{out} , Y _{out}	V _S = 3 V	1.35	1.5	1.65	V
0 g Voltage at Z _{out}	V _S = 3 V	1.2	1.5	1.8	V
0 g Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					
Noise Density X _{out} , Y _{out}			150		µg/√Hz rms
Noise Density Z _{out}			300		µg/√Hz rms
FREQUENCY RESPONSE⁴					
Bandwidth X _{out} , Y _{out} ⁵	No external filter		1600		Hz
Bandwidth Z _{out} ⁵	No external filter		550		Hz
R _{int} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X _{out}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y _{out}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z _{out}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V _S = 3 V		350		µA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S.

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).

⁵ Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S.

⁷ Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in microfarads (µF).

ADXL335

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 g
Acceleration (Any Axis, Powered)	10,000 g
V _S	-0.3 V to +3.6 V
All Other Pins	(COM = 0.3 V) to (V _S + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	-55°C to +125°C
Temperature Range (Storage)	-65°C to +150°C

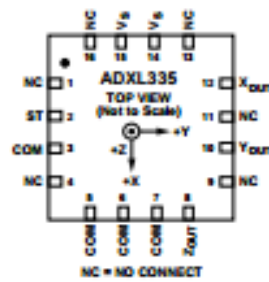
Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NC = NO CONNECT
NOTES
1. EXPOSED PAD IS NOT INTERNALLY CONNECTED BUT SHOULD BE SOLDERED FOR MECHANICAL INTEGRITY.

Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect ¹ .
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect ¹ .
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z _{out}	Z Channel Output.
9	NC	No Connect ¹ .
10	Y _{out}	Y Channel Output.
11	NC	No Connect ¹ .
12	X _{out}	X Channel Output.
13	NC	No Connect ¹ .
14	V _s	Supply Voltage (1.8 V to 3.6 V).
15	V _s	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect ¹ .
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

¹NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

ADXL335

TYPICAL PERFORMANCE CHARACTERISTICS

N > 1000 for all typical performance plots, unless otherwise noted.

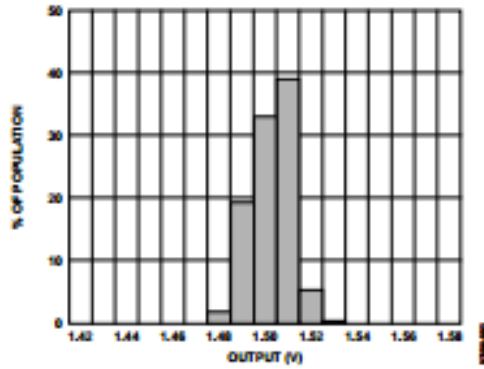


Figure 3. X-Axis Zero g Bias at 25°C, $V_s = 3\text{ V}$

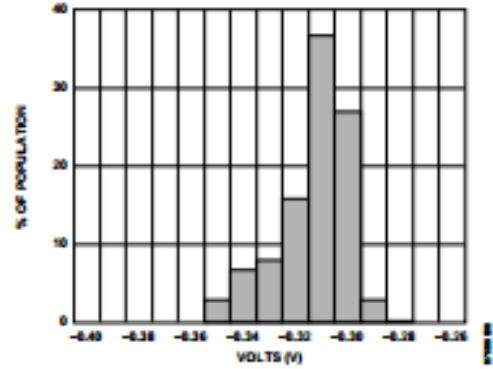


Figure 6. X-Axis Self-Test Response at 25°C, $V_s = 3\text{ V}$

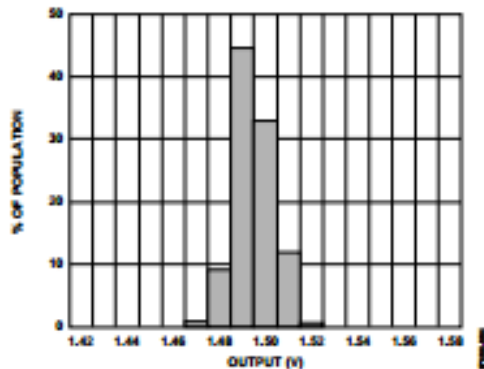


Figure 4. Y-Axis Zero g Bias at 25°C, $V_s = 3\text{ V}$

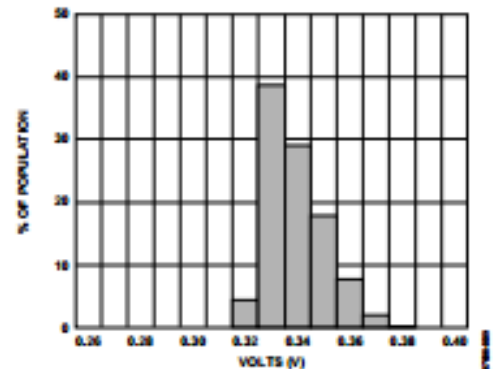


Figure 7. Y-Axis Self-Test Response at 25°C, $V_s = 3\text{ V}$

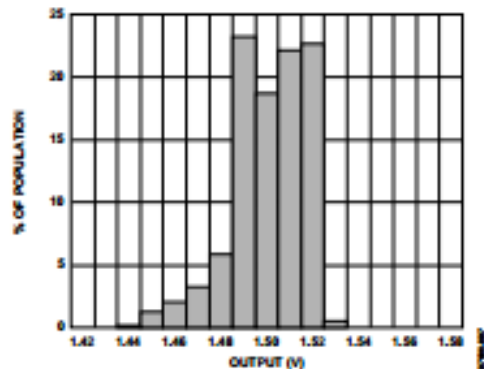


Figure 5. Z-Axis Zero g Bias at 25°C, $V_s = 3\text{ V}$

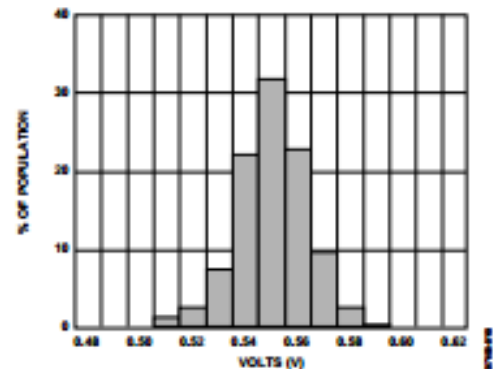


Figure 8. Z-Axis Self-Test Response at 25°C, $V_s = 3\text{ V}$

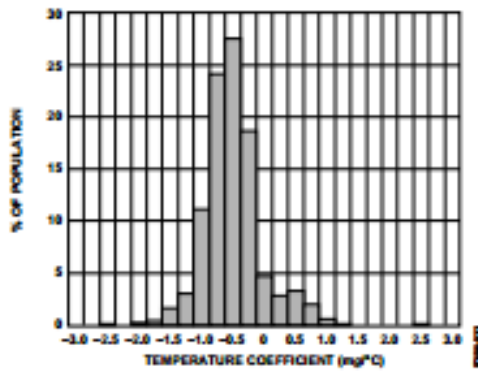


Figure 9. X-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

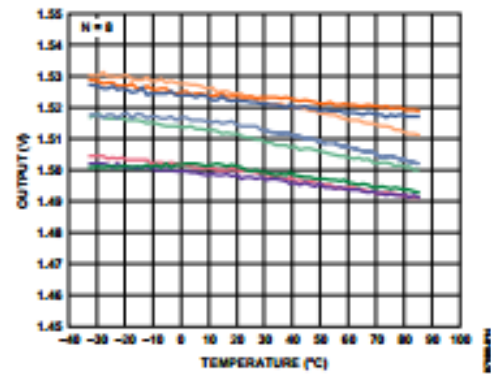


Figure 12. X-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

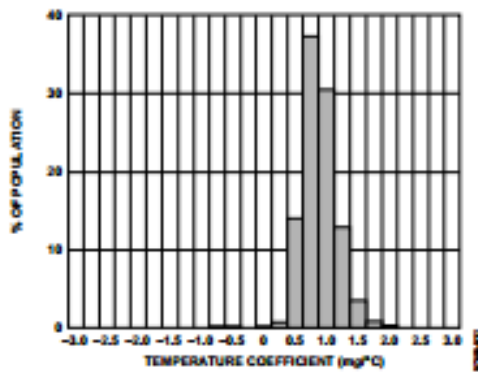


Figure 10. Y-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

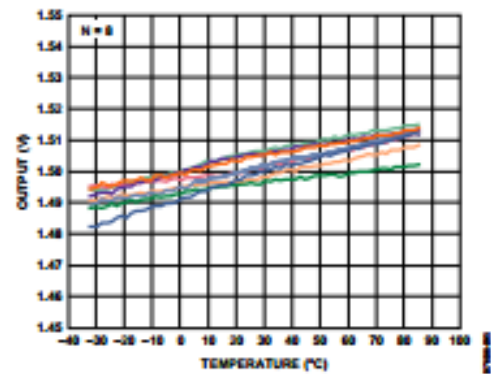


Figure 13. Y-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

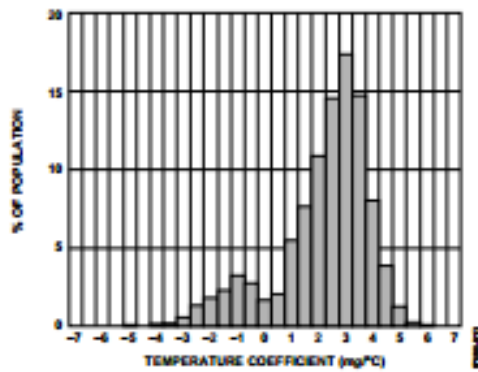


Figure 11. Z-Axis Zero g Bias Temperature Coefficient, $V_s = 3\text{ V}$

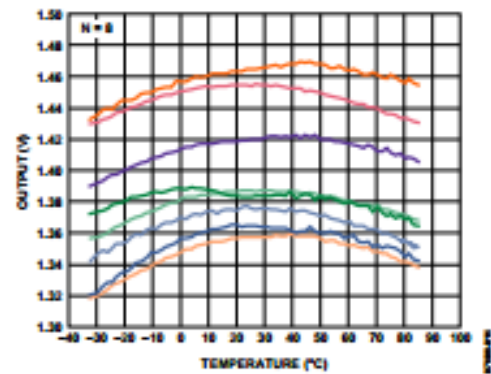


Figure 14. Z-Axis Zero g Bias vs. Temperature—
Eight Parts Soldered to PCB

ADXL335

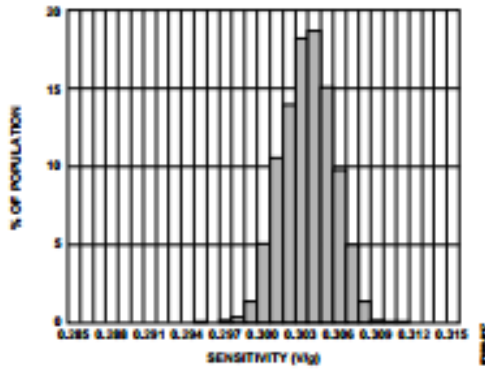


Figure 15. X-Axis Sensitivity at 25°C, $V_L = 3$ V

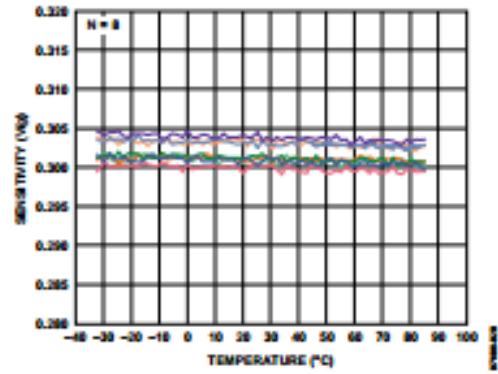


Figure 18. X-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_L = 3$ V

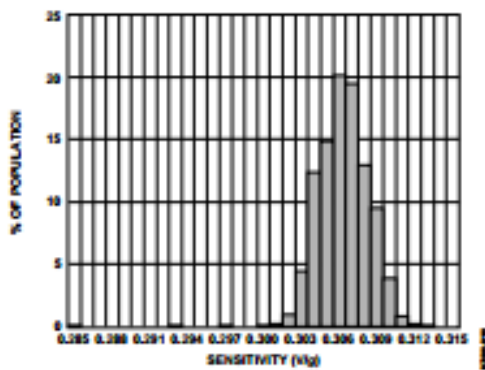


Figure 16. Y-Axis Sensitivity at 25°C, $V_L = 3$ V

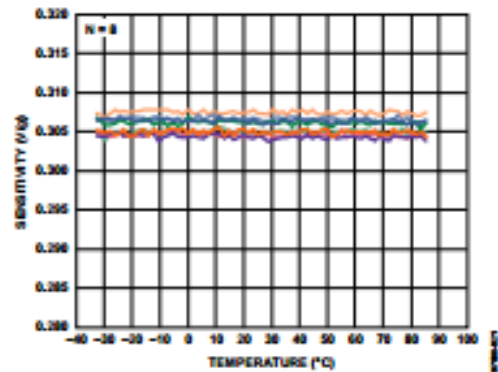


Figure 19. Y-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_L = 3$ V

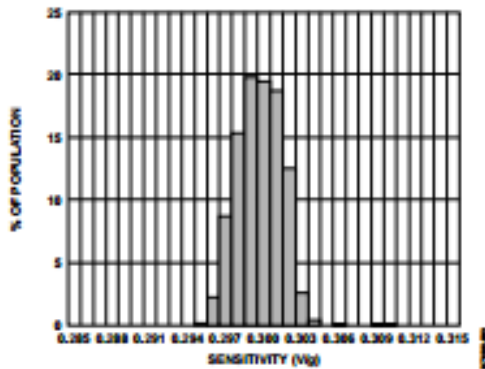


Figure 17. Z-Axis Sensitivity at 25°C, $V_L = 3$ V

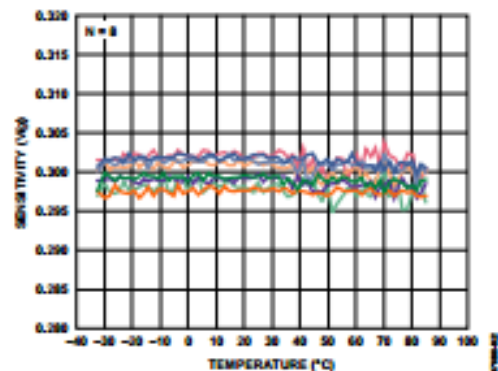


Figure 20. Z-Axis Sensitivity vs. Temperature—
Eight Parts Soldered to PCB, $V_L = 3$ V

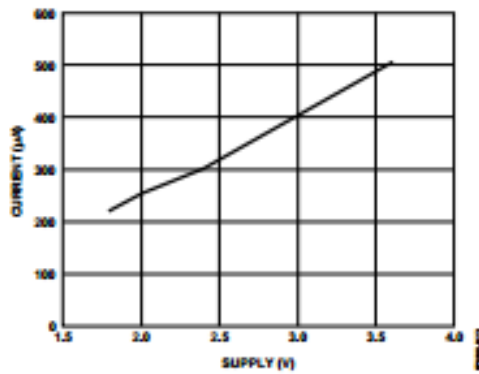


Figure 21. Typical Current Consumption vs. Supply Voltage

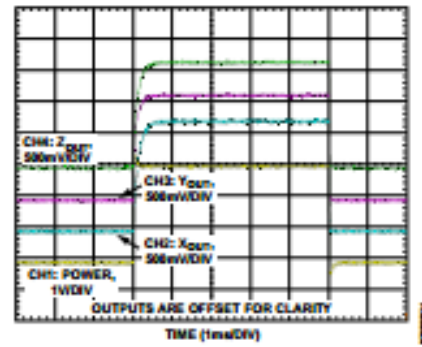


Figure 22. Typical Turn-On Time, $V_L = 3\text{ V}$

ADXL335

THEORY OF OPERATION

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of $\pm 3g$ minimum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a 32 k Ω resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

MECHANICAL SENSOR

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

PERFORMANCE

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or nonmonotonic behavior, and temperature hysteresis is very low (typically less than 3 mg over the -25°C to $+70^{\circ}\text{C}$ temperature range).

APPLICATIONS INFORMATION

POWER SUPPLY DECOUPLING

For most applications, a single 0.1 μF capacitor, C_{DEC} , placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply. However, in applications where noise is present at the 50 kHz internal clock frequency (or any harmonic thereof), additional care in power supply bypassing is required because this noise can cause errors in acceleration measurement.

If additional decoupling is needed, a 100 Ω (or smaller) resistor or ferrite bead can be inserted in the supply line. Additionally, a larger bulk bypass capacitor (1 μF or greater) can be added in parallel to C_{DEC} . Ensure that the connection from the ADXL335 ground to the power supply ground is low impedance because noise transmitted through ground has a similar effect to noise transmitted through V_{S} .

SETTING THE BANDWIDTH USING C_{X} , C_{Y} , AND C_{Z}

The ADXL335 has provisions for band limiting the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The equation for the 3 dB bandwidth is

$$F_{3\text{dB}} = 1/(2\pi(32\text{ k}\Omega) \times C_{\text{X}}) \text{ Hz}$$

or more simply

$$F_{3\text{dB}} = 5 \mu\text{F}/C_{\text{X}} \text{ Hz}$$

The tolerance of the internal resistor (R_{INT}) typically varies as much as $\pm 15\%$ of its nominal value (32 k Ω), and the bandwidth varies accordingly. A minimum capacitance of 0.0047 μF for C_{X} , C_{Y} , and C_{Z} is recommended in all cases.

Table 4. Filter Capacitor Selection, C_{X} , C_{Y} , and C_{Z}

Bandwidth (Hz)	Capacitor (μF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

SELF-TEST

The ST pin controls the self-test feature. When this pin is set to V_{S} , an electrostatic force is exerted on the accelerometer beam. The resulting movement of the beam allows the user to test if the accelerometer is functional. The typical change in output is $\pm 1.08\text{ g}$ (corresponding to $\pm 325\text{ mV}$) in the X-axis, $\pm 1.08\text{ g}$ (or $\pm 325\text{ mV}$) on the Y-axis, and $\pm 1.83\text{ g}$ (or $\pm 550\text{ mV}$) on the Z-axis. This ST pin can be left open-circuit or connected to common (COM) in normal use.

Never expose the ST pin to voltages greater than $V_{\text{S}} + 0.3\text{ V}$. If this cannot be guaranteed due to the system design (for instance, if there are multiple supply voltages), then a low V_{S} clamping diode between ST and V_{S} is recommended.

DESIGN TRADE-OFFS FOR SELECTING FILTER CHARACTERISTICS: THE NOISE/BW TRADE-OFF

The selected accelerometer bandwidth ultimately determines the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor to improve the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at X_{OUT} , Y_{OUT} , and Z_{OUT} .

The output of the ADXL335 has a typical bandwidth of greater than 500 Hz. The user must filter the signal at this point to limit aliasing errors. The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution.

The ADXL335 noise has the characteristics of white Gaussian noise, which contributes equally at all frequencies and is described in terms of $\mu\text{g}/\sqrt{\text{Hz}}$ (the noise is proportional to the square root of the accelerometer bandwidth). The user should limit bandwidth to the lowest frequency needed by the application to maximize the resolution and dynamic range of the accelerometer.

With the single-pole, roll-off characteristic, the typical noise of the ADXL335 is determined by

$$\text{rms Noise} = \text{Noise Density} \times (\sqrt{\text{BW}} \times 1.6)$$

It is often useful to know the peak value of the noise. Peak-to-peak noise can only be estimated by statistical methods. Table 5 is useful for estimating the probabilities of exceeding various peak values, given the rms value.

Table 5. Estimation of Peak-to-Peak Noise

Peak-to-Peak Value	% of Time That Noise Exceeds Nominal Peak-to-Peak Value
2 \times rms	32
4 \times rms	4.6
6 \times rms	0.27
8 \times rms	0.006

USE WITH OPERATING VOLTAGES OTHER THAN 3 V

The ADXL335 is tested and specified at $V_{\text{S}} = 3\text{ V}$; however, it can be powered with V_{S} as low as 1.8 V or as high as 3.6 V. Note that some performance parameters change as the supply voltage is varied.

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At $V_s = 3.6$ V, the output sensitivity is typically 360 mV/g. At $V_s = 2$ V, the output sensitivity is typically 195 mV/g.

The zero g bias output is also ratiometric, thus the zero g output is nominally equal to $V_s/2$ at all supply voltages.

The output noise is not ratiometric but is absolute in volts; therefore, the noise density decreases as the supply voltage increases. This is because the scale factor (mV/g) increases while the noise voltage remains constant. At $V_s = 3.6$ V, the X-axis and Y-axis noise density is typically 120 $\mu\text{g}/\sqrt{\text{Hz}}$, whereas at $V_s = 2$ V, the X-axis and Y-axis noise density is typically 270 $\mu\text{g}/\sqrt{\text{Hz}}$.

Self-test response in g is roughly proportional to the square of the supply voltage. However, when ratiometricity of sensitivity is factored in with supply voltage, the self-test response in volts is roughly proportional to the cube of the supply voltage. For example, at $V_s = 3.6$ V, the self-test response for the ADXL335 is approximately ± 560 mV for the X-axis, ± 560 mV for the Y-axis, and ± 950 mV for the Z-axis.

At $V_s = 2$ V, the self-test response is approximately ± 96 mV for the X-axis, ± 96 mV for the Y-axis, and ± 163 mV for the Z-axis.

The supply current decreases as the supply voltage decreases. Typical current consumption at $V_s = 3.6$ V is 375 μA , and typical current consumption at $V_s = 2$ V is 200 μA .

AXES OF ACCELERATION SENSITIVITY

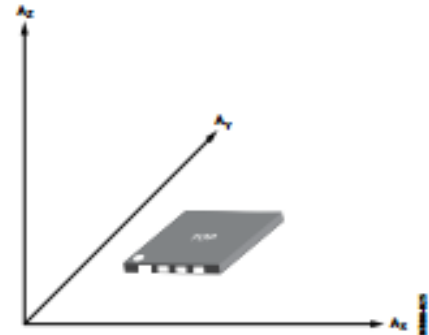


Figure 23. Axes of Acceleration Sensitivity; Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis.

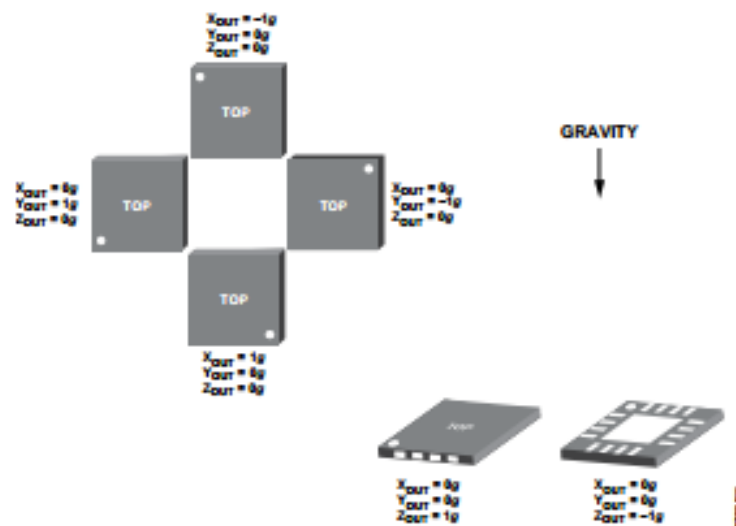


Figure 24. Output Response vs. Orientation to Gravity

LAYOUT AND DESIGN RECOMMENDATIONS

The recommended soldering profile is shown in Figure 25 followed by a description of the profile features in Table 6. The recommended PCB layout or solder land drawing is shown in Figure 26.

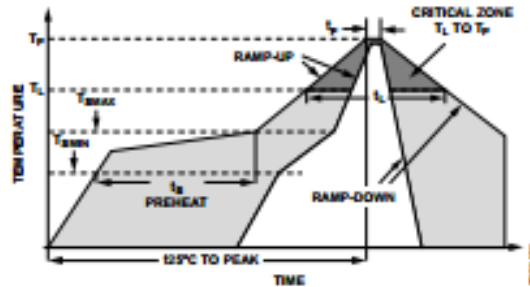


Figure 25. Recommended Soldering Profile

Table 6. Recommended Soldering Profile

Profile Feature	Sn63/Pb37	Pb-Free
Average Ramp Rate (T_L to T_P)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{MIN})	100°C	150°C
Maximum Temperature (T_{MAX})	150°C	200°C
Time (T_{MIN} to T_{MAX}) (t_2)	60 sec to 120 sec	60 sec to 180 sec
T_{MAX} to T_L		
Ramp-Up Rate	3°C/sec max	3°C/sec max
Time Maintained Above Liquidous (t_1)		
Liquidous Temperature (T_L)	183°C	217°C
Time (t_1)	60 sec to 150 sec	60 sec to 150 sec
Peak Temperature (T_P)	240°C + 0°C/-5°C	260°C + 0°C/-5°C
Time Within 5°C of Actual Peak Temperature (t_p)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

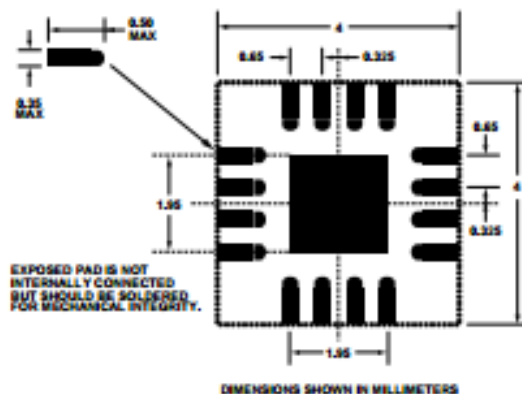


Figure 26. Recommended PCB Layout

ADXL335

OUTLINE DIMENSIONS

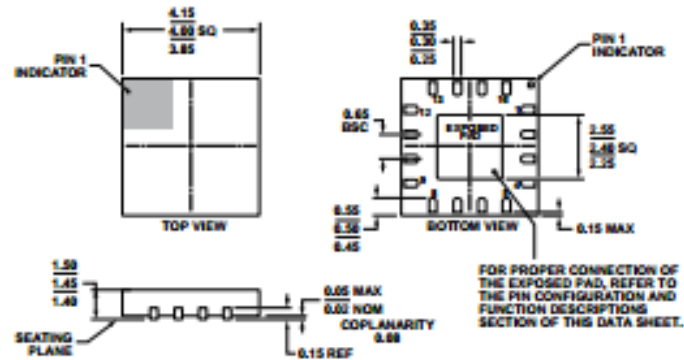


Figure 27. 16-Lead Lead Frame Chip Scale Package (LFCSP_LQ)
4 mm x 4 mm Body, 1.45 mm Thick Quad
(CP-16-14)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Measurement Range	Specified Voltage	Temperature Range	Package Description	Package Option
ADXL335BCPZ ¹	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL ¹	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
ADXL335BCPZ-RL ⁷	±3 g	3 V	-40°C to +85°C	16-Lead LFCSP_LQ	CP-16-14
EVAL-ADXL335Z ¹				Evaluation Board	

¹ Z = RoHS Compliant Part.

NOTES

ADXL335

NOTES

9.2.3 VENUS GPS



NMEA Reference Manual

SiRF Technology, Inc.
148 East Brokaw Road
San Jose, CA 95112 U.S.A.
Phone: +1 (408) 467-0410
Fax: +1 (408) 467-0420
www.SiRF.com

1050-0042
January 2005, Revision 1.3

SiRF, SiRFStar, and SiRF plus orbit design are registered in the U.S. Patent and Trademark Office. This document contains information on a product under development at SiRF. The information is intended to help you evaluate this product. SiRF reserves the right to change or discontinue work on this product without notice.

NMEA Reference Manual

Copyright © 1996-2005 SiRF Technology, Inc. All rights reserved.

No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system without the prior written permission of SiRF Technology, Inc. unless such copying is expressly permitted by United States copyright law. Address inquiries to Legal Department, SiRF Technology, Inc., 148 East Brokaw Road, San Jose, California 95112, United States of America.

About This Document

This document contains information on SiRF products. SiRF Technology, Inc. reserves the right to make changes in its products, specifications and other information at any time without notice. SiRF assumes no liability or responsibility for any claims or damages arising out of the use of this document, or from the use of integrated circuits based on this document, including, but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights. SiRF makes no warranties, either express or implied with respect to the information and specifications contained in this document. Performance characteristics listed in this data sheet do not constitute a warranty or guarantee of product performance. All terms and conditions of sale are governed by the SiRF Terms and Conditions of Sale, a copy of which you may obtain from your authorized SiRF sales representative.

Getting Help

If you have any problems contact your SiRF representative or call or send an e-mail to the SiRF Technology support group:

phone	+1 (408) 467-0410
e-mail	support@sirf.com

Figures



Tables

Table 1-1	NMEA Output Messages	1-1
Table 1-2	Supported NMEA Output Messages	1-2
Table 1-3	GGA Data Format	1-2
Table 1-4	Position Fix Indicator	1-3
Table 1-5	GLL Data Format	1-3
Table 1-6	GSA Data Format	1-4
Table 1-7	Mode 1	1-4
Table 1-8	Mode 2	1-4
Table 1-9	GSV Data Format	1-5
Table 1-10	MSS Data Format	1-5
Table 1-11	RMC Data Format	1-6
Table 1-12	VTG Data Format	1-7
Table 1-13	ZDA Data Format	1-7
Table 1-14	OkToSend Message Data Format	1-8
Table 2-1	Transport Message parameters	2-1
Table 2-2	NMEA Input Messages	2-2

Table 2-3	Supported NMEA Input Messages.....	2-2
Table 2-4	Set Serial Port Data Format.....	2-3
Table 2-5	Navigation Initialization Data Format.....	2-3
Table 2-6	Reset Configuration - Non SiRFLoc Platforms.....	2-4
Table 2-7	Reset Configuration - SiRFLoc Specific.....	2-4
Table 2-8	Set DGPS Port Data Format.....	2-4
Table 2-9	Query/Rate Control Data Format (See example 1).....	2-5
Table 2-10	Messages.....	2-5
Table 2-11	LLA Navigation Initialization Data Format.....	2-6
Table 2-12	Reset Configuration.....	2-6
Table 2-13	Development Data On/Off Data Format.....	2-7
Table 2-14	Select Datum Data Format.....	2-7
Table 2-15	RMC Data Format.....	2-8

Preface



All SiRF product support a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA).

The *NMEA Reference Manual* provides details of NMEA messages developed and defined by SiRF. It does not provide information about the complete NMEA-0183 interface standard.

Who Should Use This Guide

This manual was written assuming the user has a basic understanding of interface protocols and their use.

How This Guide Is Organized

This manual contains the following chapters:

Chapter 1, “Output Messages” defines SiRF developed NMEA output messages.

Chapter 2, “Input Messages” defines SiRF developed NMEA input messages.

Related Manuals

You can refer to the following document for more information:

- *NMEA-0183 Standard For Interfacing Marine Electronic Devices*
- *SIRF Binary Protocol Reference Manual*
- *SIRF Evaluation Kit User Guide*
- *SIRF System Development Kit User Guide*



Contacting SiRF Technical Support

Address:

SiRF Technology Inc.
148 East Brokaw Road
San Jose, CA 95112 U.S.A.

SiRF Technical Support:

Phone: +1 (408) 467-0410 (9 am to 5 pm Pacific Standard Time)

Email: support@sirf.com

General enquiries:

Phone: +1 (408) 467-0410 (9 am to 5 pm Pacific Standard Time)

Email: gps@sirf.com

Output Messages



Table 1-1 lists each of the NMEA output messages specifically developed and defined by SiRF for use within SiRF products.

Table 1-1 NMEA Output Messages

Option	Description
GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution, and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.
ZDA	PPS timing message (synchronized to PPS).
190	OK to send message.

A full description of the listed NMEA messages are provided in the following sections.

1-1

Table 1-2 provides a summary of SiRF NMEA output messages supported by the specific SiRF platforms.

Table 1-2 Supported NMEA Output Messages

Message	SiRF Software Options			
	GSW1	SiRFNTrac	SiRFLoc	GSW3
GGA	Yes	Yes	Yes	Yes
GLL	Yes	Yes	Yes	Yes
GSA	Yes	Yes	Yes	Yes
GSV	Yes	Yes	Yes	Yes
MSS	Yes	No	No	No
RMC	Yes	Yes	Yes	Yes
VTG	Yes	Yes	Yes	Yes
ZDA	2.3.2 and above	No	No	No
150	2.3.2 and above	No	No	No

Note – GSW2 software only outputs NMEA version 2.20 (and earlier). XTrac and GSW3 software have conditional defines (UI_NMEA_VERSION_XXX) to allow a choice between NMEA 2.20 and 3.00. The file `NMEA_SIRF.H` contains the NMEA version defines.

GGA—Global Positioning System Fixed Data

Note – Fields marked in *italic red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-3 contains the values for the following example:

\$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,.,.,0000*18

Table 1-3 GGA Data Format

Name	Example	Units	Description
Message ID	\$GPGGA		GGA protocol header
UTC Time	161229.487		hhmm.ss.ss
Latitude	3723.2475		ddmm.mmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

Table 1-4 Position Fix Indicator

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3-5	Not supported
<i>6</i>	<i>Dead Reckoning Mode, fix valid</i>

Note – A valid position fix indicator is derived from the SiRF Binary M.I.D. 2 position mode 1. See the *SiRF Binary Protocol Reference Manual*.

GLL—Geographic Position - Latitude/Longitude

Note – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-5 contains the values for the following example:

\$GPGLL, 3723.2475,N,12158.3416,W,161229.487,A,A*41

Table 1-5 GLL Data Format

Name	Example	Units	Description
Message ID	\$GPGLL		GLL protocol header
Latitude	3723.2475		ddmm.mmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmmm
E/W Indicator	W		E=cast or W=west
UTC Time	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR (Only present in NMEA version 3.00)</i>
Checksum	*41		
<CR> <LF>			End of message termination



GSA—GNSS DOP and Active Satellites

Note – Fields marked in *italic red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-6 contains the values for the following example:

\$GPGSA,A,3,07,02,26,27,09,04,15, 1.8,1.0,1.5*33

Table 1-6 GSA Data Format

Name	Example	Units	Description
Message ID	\$GPGSA		GSA protocol header
Mode 1	A		See Table 1-7
Mode 2	3		See Table 1-8
Satellite Used ¹	07		Sv on Channel 1
Satellite Used ¹	02		Sv on Channel 2
...			...
Satellite Used ¹			Sv on Channel 12
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		Horizontal Dilution of Precision
VDOP	1.5		Vertical Dilution of Precision
Checksum	*33		
<CR> <LF>			End of message termination

1. Satellite used in solution.

Table 1-7 Mode 1

Value	Description
M	Manual—forced to operate in 2D or 3D mode
A	2D Automatic—allowed to automatically switch 2D/3D

Table 1-8 Mode 2

Value	Description
1	Fix not available
2	2D (<4 SVs used)
3	3D (>3 SVs used)

GSV—GNSS Satellites in View

Table 1-9 contains the values for the following example:

\$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42*71

\$GPGSV,2,2,07,09,23,313,42,04,19,159,41,15,12,041,42*41

Table 1-9 GSV Data Format

Name	Example	Units	Description
Message ID	\$GPGSV		GSV protocol header
Number of Messages ¹	2		Range 1 to 3
Message Number ¹	1		Range 1 to 3
Satellites in View	07		
Satellite ID	07		Channel 1 (Range 1 to 32)
Elevation	79	degrees	Channel 1 (Maximum 90)
Azimuth	048	degrees	Channel 1 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
---			---
Satellite ID	27		Channel 4 (Range 1 to 32)
Elevation	27	degrees	Channel 4 (Maximum 90)
Azimuth	138	degrees	Channel 4 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
Checksum	*71		
<CR> <LF>			End of message termination

1. Depending on the number of satellites tracked, multiple messages of GSV data may be required.

MSS—MSK Receiver Signal

Note – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-10 contains the values for the following example:

\$GPMSS, 55,27,318.0,100,1,*57

Table 1-10 MSS Data Format

Name	Example	Units	Description
Message ID	\$GPMSS		MSS protocol header
Signal Strength	55	dB	SS of tracked frequency
Signal-to-Noise Ratio	27	dB	SNR of tracked frequency
Beacon Frequency	318.0	kHz	Currently tracked frequency
Beacon Bit Rate	100		bits per second
<i>Channel Number</i>	<i>1</i>		<i>The channel of the beacon being used if a multi-channel beacon receiver is used</i>
Checksum	*57		
<CR> <LF>			End of message termination

Note – The MSS NMEA message can only be polled or scheduled using the MSK NMEA input message. See “MSK—MSK Receiver Interface” on page 2-8.

RMC—Recommended Minimum Specific GNSS Data

Note – Fields marked in *italic red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-11 contains the values for the following example:

\$GPRMC, 161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, ,*10

Table 1-11 RMC Data Format

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC Time	161229.487		hhmmss.ss
Status ¹	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation ²		degrees	E=east or W=west
<i>Modes</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*10		
<CR> <LF>			End of message termination

1. A valid status is derived from the SIRT Binary MLD2 position mode 1. See the *SIRT Binary Protocol Reference Manual*.

2. SIRT Technology Inc. does not support magnetic declination. All “course over ground” data are geodetic WGS84 directions.

VTG—Course Over Ground and Ground Speed

Note – Fields marked in *italic red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-12 contains the values for the following example:

\$GPVTG,309.62,T,M,0.13,N,0.2,K,A*23

Table 1-12 VTG Data Format

Name	Example	Units	Description
Message ID	\$GPVTG		VTG protocol header
Course	309.62	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic ¹
Speed	0.13	knots	Measured horizontal speed
Units	N		Knots
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
Mode	A		A=Autonomous, D=DGPS, E=DR
Checksum	*23		
<CR> <LF>			End of message termination

1. SiRF Technology Inc. does not support magnetic declination. All "course over ground" data are geodetic WGS84 directions.

ZDA—SiRF Timing Message

Output: the time associated with the current 1 PPS pulse. Each message is output within a few hundred ms after the 1 PPS pulse is output and tells the time of the pulse that just occurred.

Table 1-13 contains the values for the following example:

\$GPZDA,181813,14,10,2003,00,00*4F

Table 1-13 ZDA Data Format

Name	Example	Units	Description
Message ID	\$GPZDA		ZDA protocol header
UTC time	181813		Either using valid IONOMUTC or estimated from default leap seconds
Day	14		01 TO 31
Month	10		01 TO 12
Year	2003		1980 to 2079
Local zone hour	00	knots	Offset from UTC (set to 00)
Local zone minutes	00		Offset from UTC (set to 00)
Checksum			
<CR> <LF>			End of message termination



150—OkToSend

This message is being sent out during the trickle power mode to communicate with an outside program such as SiRFDemo to indicate whether the receiver is awake or not.

Table 1-14 contains the values for the following examples:

1. OkToSend

\$PSRF150,1*3F

2. not OkToSend

\$PSRF150,0*3E

Table 1-14 OkToSend Message Data Format

Name	Example	Units	Description
Message ID	\$PSRF150		PSRF150 protocol header
OkToSend	1		1=OK to send, 0=not OK to send
Checksum	*3F		
<CR> <LF>			End of message termination

Input Messages



NMEA input messages enable you to control the Evaluation Receiver while in NMEA protocol mode. The Evaluation Receiver may be put into NMEA mode by sending the SiRF binary protocol message "Switch to NMEA Protocol - Message I.D. 129" (see the *SiRF Binary Protocol Reference Manual*). This can be done by using a user program or by using the SiRFSDemo software and selecting Switch to NMEA Protocol from the Action menu (see the *SiRF Evaluation Kit User Guide* or the *SiRFDemo User Guide*). If the receiver is in SiRF binary mode, all NMEA input messages are ignored. Once the receiver is put into NMEA mode, the following messages may be used to command the module.

Transport Message

Table 2-1 describes the transport message parameters.

Table 2-1 Transport Message parameters

Start Sequence	Payload	Checksum	End Sequence
\$PSRF<MID> ¹	Data ²	*CKSUM ³	<CR> <LF> ⁴

1. Message Identifier consisting of three numeric characters. Input messages begin at MID 100.

2. Message specific data. Refer to a specific message section for <data>...<data> definition.

3. CKSUM is a two-hex character checksum as defined in the NMEA specification, NMEA-0183 *Standard For Interfacing Marine Electronic Devices*. Use of checksums is required on all input messages.

4. Each message is terminated using Carriage Return (CR) Line Feed (LF) which is \r\n, which is hex0D 0A. Because \r\n are not printable ASCII characters, they are omitted from the example strings, but must be sent to terminate the message and cause the receiver to process that input message.

Note – All fields in all proprietary NMEA messages are required, none are optional. All NMEA messages are comma delimited.

NMEA Input Messages

Table 2-2 describes the NMEA input messages.

Table 2-2 NMEA Input Messages

Message	MTD ¹	Description
SetSerialPort	100	Set PORT A parameters and protocol
NavigationInitialization	101	Parameters required for start using X/Y/Z ²
SetDGPSPort	102	Set PORT B parameters for DGPS input
Query/Rate Control	103	Query standard NMEA message and/or set output rate
LLANavigationInitialization	104	Parameters required for start using Lat/Lon/Air ³
Development Data On/Off	105	Development Data messages On/Off
Select Datum	106	Selection of datum to be used for coordinate transformations.
MSK Receiver Interface	MSK	Command message to a MSK radio-beacon receiver.

1. Message Identification (MTD).

2. Input coordinates must be WGS84.

3. Input coordinates must be WGS84.

Note – NMEA input messages 100 to 106 are SiRF proprietary NMEA messages. The MSK NMEA string is as defined by the NMEA 0183 standard.

Table 2-3 provides a summary of supported SiRF NMEA input messages by the specific SiRF platforms.

Table 2-3 Supported NMEA Input Messages

Message ID	SiRF Software Options		
	GSW1	SiRFNTrac	SiRFLoc
100	Yes	Yes	Yes
101	Yes	No	Yes
102	Yes	No	No
103	Yes	Yes	Yes
104	Yes	No	Yes
105	Yes	Yes	Yes
106	Yes	Yes	Yes
MSK	Yes	No	No

100—SetSerialPort

This command message is used to set the protocol (SiRF binary or NMEA) and/or the communication parameters (Baud, data bits, stop bits, and parity). Generally, this command is used to switch the module back to SiRF binary protocol mode where a more extensive command message set is available. When a valid message is received, the parameters are stored in battery-backed SRAM and the Evaluation Receiver restarts using the saved parameters.

Table 2-4 contains the input values for the following example:

Switch to SiRF binary protocol at 9600,8,N,1

\$PSRF100,0,9600,8,1,0*0C

Table 2-4 Set Serial Port Data Format

Name	Example	Units	Description
Message ID	\$PSRF100		PSRF100 protocol header
Protocol	0		0=SiRF binary, 1=NMEA
Baud	9600		4800, 9600, 19200, 38400
DataBits	8		8,7 ¹
StopBits	1		0,1
Parity	0		0=None, 1=Odd, 2=Even
Checksum	*0C		
<CR> <LF>			End of message termination

1. SiRF protocol is only valid for 8 data bits, 1 stop bit, and no parity.

101—NavigationInitialization

This command is used to initialize the Evaluation Receiver by providing current position (in X, Y, Z coordinates), clock offset, and time. This enables the Evaluation Receiver to search for the correct satellite signals at the correct signal parameters. Correct initialization parameters enable the Evaluation Receiver to acquire signals quickly.

Table 2-5 contains the input values for the following example:

Start using known position and time.

\$PSRF101,-2686700,-4304200,3851624,96000,497260,921,12,3*1C

Table 2-5 Navigation Initialization Data Format

Name	Example	Units	Description
Message ID	\$PSRF101		PSRF101 protocol header
ECEF X	-2686700	meters	X coordinate position
ECEF Y	-4304200	meters	Y coordinate position
ECEF Z	3851624	meters	Z coordinate position
ClkOffset	96000	Hz	Clock Offset of the Evaluation Receiver ¹
TimeOfWeek	497260	seconds	GPS Time Of Week
WeekNo	921		GPS Week Number
ChannelCount	12		Range 1 to 12
ResetCfg	3		See Table 2-6 and Table 2-7
Checksum	*1C		
<CR> <LF>			End of message termination

1. Use 0 for last saved value if available. If this is unavailable, a default value of 96,000 is used.

Table 2-6 Reset Configuration - Non SiRFLoc Platforms

Hex	Description
0x01	Hot Start—All data valid
0x02	Warm Start—Ephemeris cleared
0x03	Warm Start (with Init)—Ephemeris cleared, initialization data loaded
0x04	Cold Start—Clears all data in memory
0x08	Clear Memory—Clears all data in memory and resets the receiver back to factory defaults

Table 2-7 Reset Configuration - SiRFLoc Specific

Hex	Description
0x00	Perform a hot start using internal RAM data. No initialization data is used.
0x01	Use initialization data and begin in start mode. Uncertainties are 5 seconds time accuracy and 300 km position accuracy. Ephemeris data in SRAM is used.
0x02	No initialization data is used, ephemeris data is cleared, and warm start performed using remaining data in RAM.
0x03	Initialization data is used, ephemeris data is cleared, and warm start performed using remaining data in RAM.
0x04	No initialization data is used. Position, time and ephemeris are cleared and a cold start is performed.
0x08	No initialization data is used. Internal RAM is cleared and a factory reset is performed.

102—SetDGPSPort

This command is used to control the serial port used to receive RTCM differential corrections. Differential receivers may output corrections using different communication parameters. If a DGPS receiver is used that has different communication parameters, use this command to allow the receiver to correctly decode the data. When a valid message is received, the parameters are stored in battery-backed SRAM and the receiver restarts using the saved parameters.

Table 2-8 contains the input values for the following example:

Set DGPS Port to be 9600,8,N,1.

\$PSRF102,9600,8,1,0*12

Table 2-8 Set DGPS Port Data Format

Name	Example	Units	Description
Message ID	\$PSRF102		PSRF102 protocol header
Baud	9600		4800, 9600, 19200, 38400
DataBits	8		8,7
StopBits	1		0,1
Parity	0		0=None, 1=Odd, 2=Even
Checksum	*12		
<CR> <LF>			End of message termination

103—Query/Rate Control

This command is used to control the output of standard NMEA messages GGA, GLL, GSA, GSV, RMC, and VTG. Using this command message, standard NMEA messages may be polled once, or setup for periodic output. Checksums may also be enabled or disabled depending on the needs of the receiving program. NMEA message settings are saved in battery-backed memory for each entry when the message is accepted.

Table 2-9 contains the input values for the following examples:

1. Query the GGA message with checksum enabled

\$PSRF103,00,01,00,01*25

2. Enable VTG message for a 1 Hz constant output with checksum enabled

\$PSRF103,05,00,01,01*20

3. Disable VTG message

\$PSRF103,05,00,00,01*21

Table 2-9 Query/Rate Control Data Format (See example 1)

Name	Example	Units	Description
Message ID	\$PSRF103		PSRF103 protocol header
Msg	00		See Table 2-10
Mode	01		0=SetRate, 1=Query
Rate	00	seconds	Output—off=0, max=255
Checksum Enable	01		0=Disable Checksum, 1=Enable Checksum
Checksum	*25		
<CR> <LF>			End of message termination

Table 2-10 Messages

Value	Description
0	GGA
1	GLL
2	GSA
3	GSV
4	RMC
5	VTG
6	MSS (if internal beacon is supported)
7	Not defined
8	ZDA (if 1PPS output is supported)
9	Not defined

Note – In TricklePower mode, update rate is specified by the user. When switching to NMEA protocol, the message update rate is also required. The resulting update rate is the product of the TricklePower Update rate and the NMEA update rate (i.e., TricklePower update rate = 2 seconds, NMEA update rate = 5 seconds, resulting update rate is every 10 seconds, (2 X 5 = 10)).

104—LLANavigationInitialization

This command is used to initialize the Evaluation Receiver by providing current position (in latitude, longitude, and altitude coordinates), clock offset, and time. This enables the receiver to search for the correct satellite signals at the correct signal parameters. Correct initialization parameters enable the receiver to acquire signals quickly.

Table 2-11 contains the input values for the following example:

Start using known position and time.

\$PSRF104,37.3875111,-121.97232,0,96000,237759,1946,12,1*07

Table 2-11 LLA Navigation Initialization Data Format

Name	Example	Units	Description
Message ID	\$PSRF104		PSRF104 protocol header
Lat	37.3875111	degrees	Latitude position (Range 90 to -90)
Lon	-121.97232	degrees	Longitude position (Range 180 to -180)
Alt	0	meters	Altitude position
ClkOffset	96000	Hz	Clock Offset of the Evaluation Receiver ¹
TimeOfWeek	237759	seconds	GPS Time Of Week
WeekNo	1946		Extended GPS Week Number (1024 added)
ChannelCount	12		Range 1 to 12
ResetCfg	1		See Table 2-12
Checksum	*07		
<CR> <LF>			End of message termination

1. Use 0 for last used value if available. If this is unavailable, a default value of 96,000 is used.

Table 2-12 Reset Configuration

Hex	Description
0x01	Hot Start—All data valid
0x02	Warm Start—Ephemeris cleared
0x03	Warm Start (with Init)—Ephemeris cleared, initialization data loaded
0x04	Cold Start—Clears all data in memory
0x08	Clear Memory—Clears all data in memory and resets receiver back to factory defaults

105—Development Data On/Off

Use this command to enable development data information if you are having trouble getting commands accepted. Invalid commands generate debug information that enables the you to determine the source of the command rejection. Common reasons for input command rejection are invalid checksum or parameter out of specified range.

Table 2-13 contains the input values for the following examples:

1. Debug On

\$PSRF105,1*3E

2. Debug Off

\$PSRF105,0*3F

Table 2-13 Development Data On/Off Data Format

Name	Example	Units	Description
Message ID	\$PSRF105		PSRF105 protocol header
Debug	1		0=Off, 1=On
Checksum	*3E		
<CR> <LF>			End of message termination

106—Select Datum

GPS receivers perform initial position and velocity calculations using an earth-centered earth-fixed (ECEF) coordinate system. Results may be converted to an earth model (geoid) defined by the selected datum. The default datum is WGS 84 (World Geodetic System 1984) which provides a worldwide common grid system that may be translated into local coordinate systems or map datums. (Local map datums are a best fit to the local shape of the earth and not valid worldwide.)

Table 2-14 contains the input values for the following examples:

1. Datum select TOKYO_MEAN

\$PSRF106,178*32

Table 2-14 Select Datum Data Format

Name	Example	Units	Description
Message ID	\$PSRF106		PSRF106 protocol header
Datum	178		21=WGS84 178=TOKYO_MEAN 179=TOKYO_JAPAN 180=TOKYO_KOREA 181=TOKYO_OKINAWA
Checksum	*32		
<CR> <LF>			End of message termination

MSK—MSK Receiver Interface

Table 2-15 contains the values for the following example:

\$GPMSK,318.0,A,100,M,2,*45

Table 2-15 RMC Data Format

Name	Example	Units	Description
Message ID	\$GPMSK		MSK protocol header
Beacon Frequency	318.0	kHz	Frequency to use
Auto/Manual Frequency ¹	A		A : Auto, M : Manual
Beacon Bit Rate	100		Bits per second
Auto/Manual Bit Rate ¹	M		A : Auto, M : Manual
Interval for Sending \$-MSS ²	2	sec	Sending of MSS messages for status

1. If Auto is specified the previous field value is ignored.

2. When status data is not to be transmitted this field is null.

Note – The NMEA messages supported by the Evaluation Receiver does not provide the ability to change the DGPS source. If you need to change the DGPS source to internal beacon, use the SiRF binary protocol and then switch to NMEA.



ADDITIONAL AVAILABLE PRODUCT INFORMATION

Part Number	Description
1050-0042	NMEA Reference Manual
1050-0041	SIRF Binary Protocol Reference Manual
1055-0136	Product Inserts
1050-0058	SIRFstarIII System Development Kit User Guide
1050-0053	GSW3 Software System Development Kit Reference Manual
1050-0054	S38DK Board System Development Kit Reference Manual
1050-0055	GSF3 Chip System Development Kit Reference Manual
1055-1034	GSF3T Data Sheet
1055-1035	GSF3w Data Sheet
	Available on the Developer Web Site
APNT3001	SSII System Guidelines and Considerations
APNT3002	PCB Design Guidelines for SSII Implementations
APNT3003	Back-Up Power Operation for SSII Architectures
APNT3004	Troubleshooting Notes for SSII Board Development
APNT3005	Co-Location and Jamming Considerations for SSII Integration
APNT3006	GPIO Pin Functionality for SSII
APNT3007	IO Message Definitions for SSII
APNT3008	Implementing User Tasks in the SSII Architecture
APNT3009	Effects of User Tasks on GPS Performance for SSII
APNT3010	Advanced Power Management (APM) Considerations for SSII
APNT3011	Multitask Testing Issues for SSII
APNT3012	Production Testing of SSII Modules
APNT3014	Automotive Design Considerations for SSII

SIRF Technology Inc.

148 East Brokaw
San Jose, CA 95112
Tel: +1-408-467-0410
Fax: +1-408-467-0420
Email: gsa@sirf.com
Website: <http://www.sirf.com>

SIRF Texas

Tel: +1-972-236-6968
Fax: +1-972-236-0372
Email: SalesAmericas@sirf.com

SIRF United Kingdom

Tel: +44-1344-668390
Fax: +44-1344-668157
Email: SalesUK@sirf.com

SIRF Japan

Tel: +81 44829-2188
Fax: +81 44829-2187
Email: SalesJapan@sirf.com

SIRF France

Tel: +33-6-0717-7882
Fax: +44-1344-668157
Email: SalesFrance@sirf.com

SIRF Germany

Tel: +49-81-529932-90
Fax: +49-81-529931-70
Email: SalesGermany@sirf.com

SIRF Taiwan

Tel: +886-2-2723-7853
Fax: +886-2-2723-7854
Email: SalesAsiaPacific@sirf.com

SIRF India

Tel: +91-120-251-0256
Fax: +91-120-251-0584
Email: SalesIndia@sirf.com

NMEA Reference Manual

© 2005 SIRF Technology Inc. All rights reserved.

Products made, sold or licensed by SIRF Technology, Inc. are protected by one or more of the following United States patents: 5,488,379; 5,504,482; 5,552,754; 5,552,382; 5,536,077; 5,593,595; 5,597,505; 5,591,171; 5,517,303; 5,523,263; 5,016,704; 5,037,900; 5,041,280; 5,044,105; 5,047,017; 5,051,228; 5,114,952; 5,125,325; 5,159,755; 5,236,337; 5,249,542; 5,278,403; 5,282,211; 5,282,749; 5,297,771; 5,301,545; 5,304,219; 5,351,485; 5,351,711; 5,355,250; 5,359,251; 5,383,046; 5,400,753; 5,421,633; 5,427,120; 5,427,121; 5,453,236; and AU728,697.

Other United States and foreign patents are issued or pending. SIRF, SIRFStar, SIRF plus, Orbit design are registered in the U.S. Patent and Trademark office. SnapLock, SnapGet, SingleGet, Foliage Lock, TricklePower, Push-to-Fix, WinSIRF, SIRFLoc, SIRFDave, SIRFNav, SIRFXtra, SIRFSoft, SoftGPS, UrbanGPS, and Multitask Location Engine are trademarks of SIRF Technology, Inc. Other trademarks are property of their respective companies.

This document contains information on SIRF products. SIRF reserves the right to make changes in its products, specifications and other information at any time without notice. SIRF assumes no liability or responsibility for any claims or damages arising out of the use of this document, or from the use of integrated circuits based on this data sheet, including, but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights. No license, either expressed or implied, is granted to any intellectual property rights of SIRF. SIRF makes no warranties, either express or implied with respect to the information and specification contained in this document. Performance characteristics listed in this document do not constitute a warranty or guarantee of product performance. SIRF products are not intended for use in life support systems or for life saving applications. All terms and conditions of sale are governed by the SIRF Terms and Conditions of Sale, a copy of which may obtain from your authorized SIRF sales representative.

December 2004



Please
Recycle